REPOR' NO. GDC-DBB70-002
CONTRACT NAS 9-11193

C R 114986

SPACE SHUTTLE SYNTHESIS PROGRAM (SSSP)

VOLUME I, PART 1 • ENGINEERING AND PROGRAMMING DISCUSSION

FINAL REPORT

GENERAL DYNAMICS
Convair Aerospace Division

REPORT NO. GDC-DBB70-002

SPACE SHUTTLE SYNTHESIS PROGRAM (SSSP)

VOLUME I, PART 1 • ENGINEERING AND PROGRAMMING DISCUSSION

FINAL REPORT

December 1970

Prepared Under
Contract NAS9-11193

Submitted to
National Aeronautics and Space Administration
MANNED SPACECRAFT CENTER
Houston, Texas

Prepared by
CONVAIR AEROSPACE DIVISION OF GENERAL DYNAMICS
San Diego, California

FOREWORD

The SSSP documentation is presented in two volumes. Volume I contains the basic user's manual text and all of the simulation input and output descriptions as well as a complete listing of the computer program FORTRAN V source deck. Volume II contains a compilation of statistical data on previous aircraft, missiles and space systems to serve as background information and program inputs to the weight/volume portion of the program.

This report is the first of three documents for Volume I. Part 2 contains the program operating instructions and Part 3 describes the program output and contains all of the Volume I appendices.

SUMMARY

The Space Shuttle Synthesis Program (SSSP) automates the trajectory, weights and performance computations essential to predesign of the Space Shuttle system for earth-to-orbit operations. The two-stage Space Shuttle system is a completely reusable space transportation system consisting of a booster and an orbiter element. The SSSP's major parts are a detailed weight/volume routine, a precision three-dimensional trajectory simulation, and the iteration and synthesis logic necessary to satisfy the hardware and trajectory constraints.

The SSSP is a highly useful tool in conceptual design studies where the effects of various trajectory configuration and shuttle subsystem parameters must be evaluated relatively rapidly and economically. The program furnishes sensitivity and tradeoff data for proper selection of configuration and trajectory predesign parameters. Emphasis is placed upon predesign simplicity and minimum input preparation. Characteristic equations for describing aerodynamic and propulsion models and for computing weights and volumes are kept relatively simple. The synthesis program is designed for a relatively large number of two-stage Space Shuttle configurations and mission types, but avoids the complexity of a completely generalized computer program that would be unwieldy to use and/or modify.

CONTENTS

LIST OF FIGURES

List of Figures (continued)

List of Figures (continued)

LIST OF TABLES

1.0 INTRODUCTION

This report provides an engineering discussion and user's document for the Space Shuttle Synthesis Program (SSSP). The SSSP has been developed to provide a fast but moderately detailed and accurate tool for Space Shuttle launch vehicle sizing and performance evaluation. The two-stage Space Shuttle system is a completely reusable space transportation system consisting of a booster and an orbiter element. Several preliminary, predesign studies of the system have previously been performed, yielding baseline configurations, sensitivities and some nominal missions and groundrules. This data bank of information was utilized as a basis for the development of the SSSP.

The SSSP couples weight/volume and trajectory computations in one computer program and provides a proper interface between vehicle design parameters and the trajectory throughout trade studies for two-stage fully recoverable Space Shuttle systems. This program was assembled by using the subroutine building-block approach with the two major subprograms (weight/volume and trajectory) and with the executive program which acts as the interface between the two major subprograms. The executive program, or synthesis driver, provides the iteration logic and adjustments required for consistency among the weight, volume, geometry and trajectory data and mission constraints. The result is a useful, economical special-purpose predesign program, which yields trajectory histories and vehicle descriptions with considerable detail and with accuracy sufficient for generating reliable sensitivity data.

This report discusses the development, structure and usage of SSSP. In Section 2.0 the overall organization, engineering development, program options, and operating modes are described. Section 3.0 contains a discussion of the operational procedures for the program along with a discussion of each subroutine. Section 4.0 describes the data-deck set up and input parameters. Section 5.0 discusses the output blocks of data and presents a sample case output. Volume II of this report contains a detailed engineering description of the weight/volume portion of the computer program along with a compilation of statistical data from which to draw initial input.

2.0 ENGINEERING DISCUSSION

This section describes the Space Shuttle mission and vehicle model, discusses the SSSP rationale, groundrules, methodology and computational flow, and presents some guidelines for efficient program usage.

2.1 MISSION AND VEHICLE MODEL

2.1.1 <u>The Space Shuttle Concept.</u> The two-stage Space Shuttle concept is shown in Figure 2-1. The two vehicles are mated and launched vertically with the orbiter element attached in a piggy-back fashion on the booster element. A typical mission is logistics resupply of an orbital space station.

During the boost phase usually only the booster element engines are burning. At staging, when the booster element has depleted its main propellants, the stages separate and the booster performs a glide/decelerate maneuver to subsonic velocity where the turbojet engines are started and cruiseback is initiated for a conventional airplane type landing at an airfield usually in the proximity of the launch site. Subsonic cruise range to the launch site is about 400 nautical miles for a typical mission. After staging the orbital element engines are ignited and the stage accelerates to orbit, docks at the space station and transfers passengers and cargo to the station. Any passengers or equipment for return to earth are then loaded aboard the vehicle. A gliding/maneuvering entry into the earth's atmosphere is made so that the vehicle arrives over the landing site. Turbojet engines are ignited and the vehicle makes a conventional airplane type landing. Following visual inspection and minimum refurbishment, the two stages are ready for payload integration, vertical assembly, refueling and relaunch.

Because the Space Shuttle concept employs many common hardware items and features for both stages such as main engines and structure concepts, Space

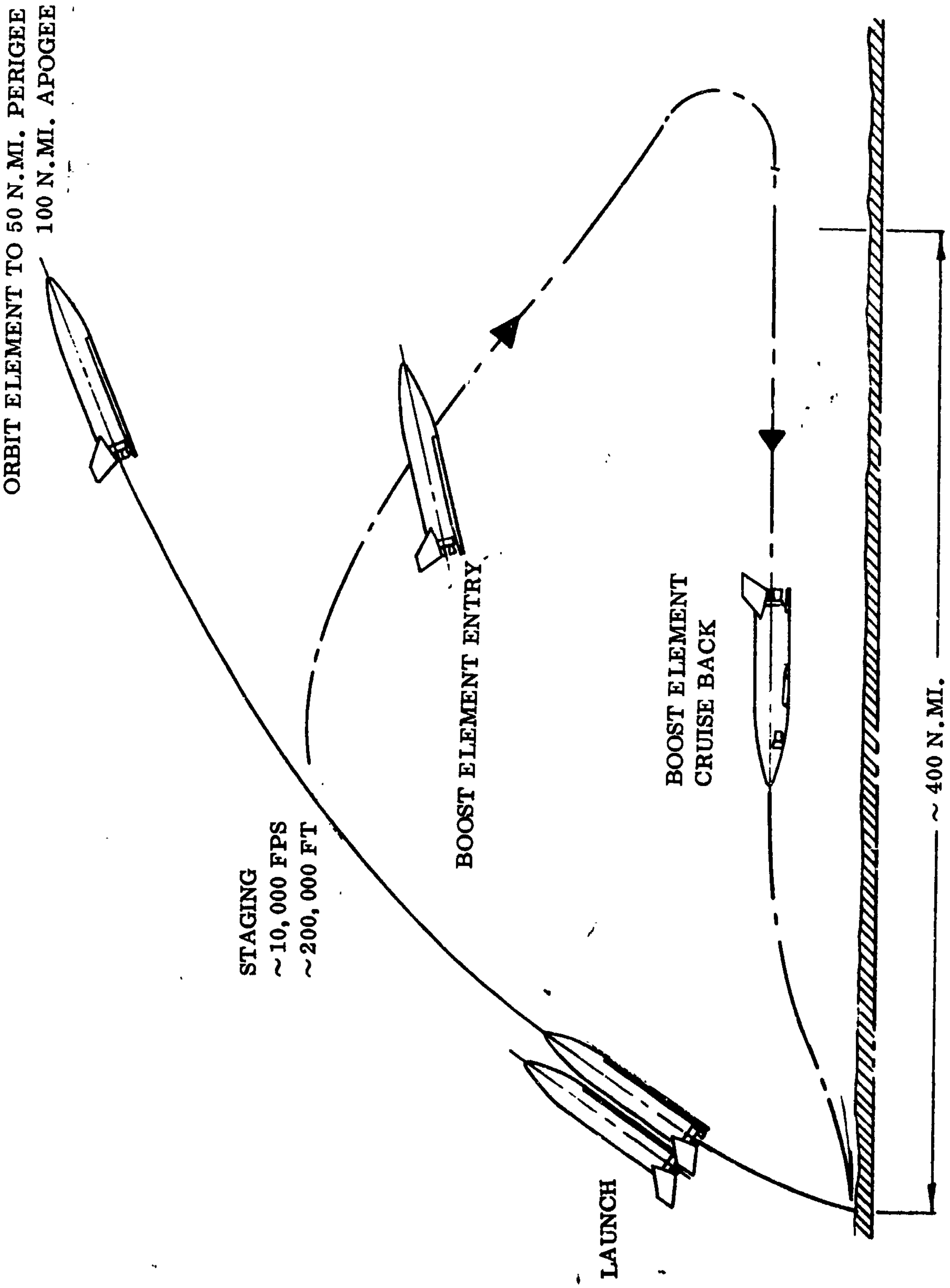


Figure 2-1. Space Shuttle System

Shuttle has the potential of greatly reduced RLT & E cost. Since there are no throw away stages, Space Shuttle also has a low recurring cost per launch. In addition, Space Shuttle has the potential of unrestricted launch azimuth capability. Payload growth and/or high altitude orbit payloads can be accommodated by using existing expendable stages such as S-IVB, Agena, Centaur, etc.

2.1.2 Trajectory Profile. To provide a common basis for predesign studies and to simplify the SSSP development and usage, a baseline ascent trajectory profile was established. The terminal conditions for the ascent trajectory are perigee injection into an elliptic parking orbit (usually 50 n. mi. perigee altitude with 100 n. mi. apogee altitude). This parking orbit provides a reasonable start for space station logistics and other missions. Insertion at this low altitude provides good performance and allows an efficient entry trajectory for the booster stage. (The orbiter entry trajectory is initiated by retro from orbit and is therefore not dependent on the ascent trajectory).

The ascent trajectory sequence is as follows:

1. Vertical rise for a specified time ($\approx$8-10 seconds)
2. Pitchover ($\approx$16 seconds)
3. Gravity turn ($\alpha = 0$ between thrust and velocity vector) maneuver to booster propellant depletion, stage separation (booster entry initiated)
4. Orbiter burn with linear cotangent steering ($\cot \psi = A+Bt$) to perigee insertion

Seven basic flight simulation sections are available to model this ascent trajectory profile. A multiplier on the pitch rate during the initial pitchover maneuver is iteratively determined to yield a specified flight path angle (or dynamic pressure) at stage separation. This separation flight path angle (or dynamic pressure) is chosen to yield a near-optimal ascent trajectory, a "cool" booster entry trajectory with short cruise range requirements, and an acceptable environment for stage separation if necessary.

The orbiter flight is then simulated with the two parameters A and B (the cotangent of the pitch attitude being linear in time) being determined to yield specified injection altitude (h_f) and injection flight path angle (γ_f) at attainment of the specified injection velocity (V_f). (The weight iteration necessary to make propellant expended by the orbiter to achieve V_f agree with the weight-sizing propellant computations is discussed later). This simplified pitch control program yields near-optimal performance for a wide variety of vehicle parameters and yields good convergence properties for the trajectory iterations.

At stage separation the trajectory conditions (V_s, h_s, γ_s, etc) are stored for use in determining the cruise fuel requirements of the booster stage. This determination may be accomplished with a number of program options; all of which are described in detail later and are based on the cruise range requirement for the mission and Breguet's equation for the fuel required for a constant - L/D cruise. Subsonic L/D and specific fuel consumption are input constants. The following lists the options available for determination of the booster cruise requirements:

1. Stored empirical data for flyback range to the launch site as functions of the trajectory conditions at stage separation and a constant angle-of-attack entry profile.

2. Flyback range to the launch site as a function of the dynamic pressure at stage separation.

3. Constant cruise range requirement.

4. Cruise range requirement as determined by the sum of the range from liftoff to staging and the instantaneous impact range (conic solution) from staging.
5. Cruise range required via numerical integration of the equations of motion for the booster entry to start of subsonic cruise to specified landing site.

The booster entry trajectory is therefore not numerically integrated except for the last option.

The last option for determining the booster stage cruise fuel requirements is entirely user controlled via input and the modeling options available as opposed to the ascent flight profile which is closely coupled to the weight-sizing iterations and is therefore constrained to definite modeling control. Eight flight simulation sections are available to model this entry trajectory profile. The landing site may be selected with input coordinates or return to the launch site as a built-in default option. The booster subsonic cruise flight may be modeled to account for transition range increments, idle descent cruise fuel losses and final descent fuel losses.

2.1.3 Vehicle Characteristics and Constraints. The fundamental concept of Space Shuttle is the complete reusability of both stages with the maximum use of such common hardware items as the main rocket engines. The booster and orbiter engines are essentially the same, although a larger number of engines will be installed on the booster than on the orbiter (e.g., 12 booster engines and 2 orbiter engines) and an extendable skirt may be added to the orbiter nozzle to improve vacuum performance. The computation sequence in the SSSP was chosen to best provide this propulsion system commonality. SSSP input specifies the ratio of the booster to orbiter vacuum thrust, T_b/T_o, the number engines per stage, and the thrust and the specific impulses (vacuum and sea level) of each type.

Man-rating the vehicle for a wide variety of possible passenger types imposes a limit on loads (probably 3 to 4 g's). This requires throttling of the rocket engines during the main burn after a specified axial load is reached. SSSP provides this option; maximum loads are specified separately for the booster and orbiter flight phases, and the engines are throttled to hold the axial loads constant after the limit is attained.

Structural, dynamic and thermodynamic constraints (such as maximum α q loading, balance, heating, etc.) are not considered in the SSSP. The effects of these constraints are analyzed externally by monitoring and using SSSP trajectory, weights and geometry data.

The SSSP provides for a number of basic options which may be utilized to constrain the basic vehicle design or to investigate alternative approaches to the Space Shuttle concept. The following lists the major options provided by input control:

1. Fixed Payload. Solve for the gross liftoff weight (GLOW).
2. Fixed GLOW. Solve for the payload.
3. Fixed Orbiter Gross Weight. Solve for the payload or GLOW.
4. Fixed Orbiter Main Propellant. Solve for payload or GLOW.
5. Fixed Liftoff Thrust/Weight (T/W_{LO}). Solve for the GLOW.

The first four options assume fixed-size engines (may constrain thrust levels) in both the booster and the orbiter stages whereas the fifth option may be used in one of two modes: (1) common engines in both the booster and orbiter, or (2) "rubber" size booster engines and fixed-size orbiter engines. In both cases the booster thrust level is defined by the specified liftoff thrust/weight and the resulting GLOW.

The SSSP also provides for secondary options which were developed during the early phases of examining the Space Shuttle concept. The following lists these secondary options provided by input control:

1. Fixed-size Solid Rocket Motor Strap-ons. These motors are utilized during the booster burn for thrust augmentation at liftoff and are jettisoned prior to booster fuel depletion.

2 Simultaneous Burn of Stages. The SSSP assumes the booster and orbiter engines are firing from liftoff to staging of the booster element. (Note, the major options listed above assumed that the two stages were sequentially burned). In this case the orbiter rocket engines may be fed from the booster propellant tanks or from the orbiter tanks, i.e., a crossfeed or no-crossfeed mode is available.

2.2 SYNTHESIS METHODOLOGY

The SSSP is designed strictly for use in predesign, with layout drawings of baseline vehicles and separate analyses of dynamics and thermodynamics as essential companions. Minor design changes (or variations about a point design) may not require new drawings; SSSP's primary job is to determine the implications of the design change on system performance, gross weights, etc., consistent with vehicle and mission constraints. For such minor changes, weight, volume and geometry sizing equations can (and should) be relatively simple, utilizing such procedures as similar-body scaling wherever possible. Major changes may require new drawings, major redesign, and an essentially new set of baseline data for SSSP. The program is thus not self-sufficient but is intended to reduce the work and time required for the predesign cycle, from point design to new point design, by providing fast, reliable sensitivity data. Figure 2-2 illustrates this process and shows the interrelationship between the predesign analyses and the SSSP.

The methodology and types of equations used in the major subprograms (weight/volume and trajectory) are described in this section. The next section

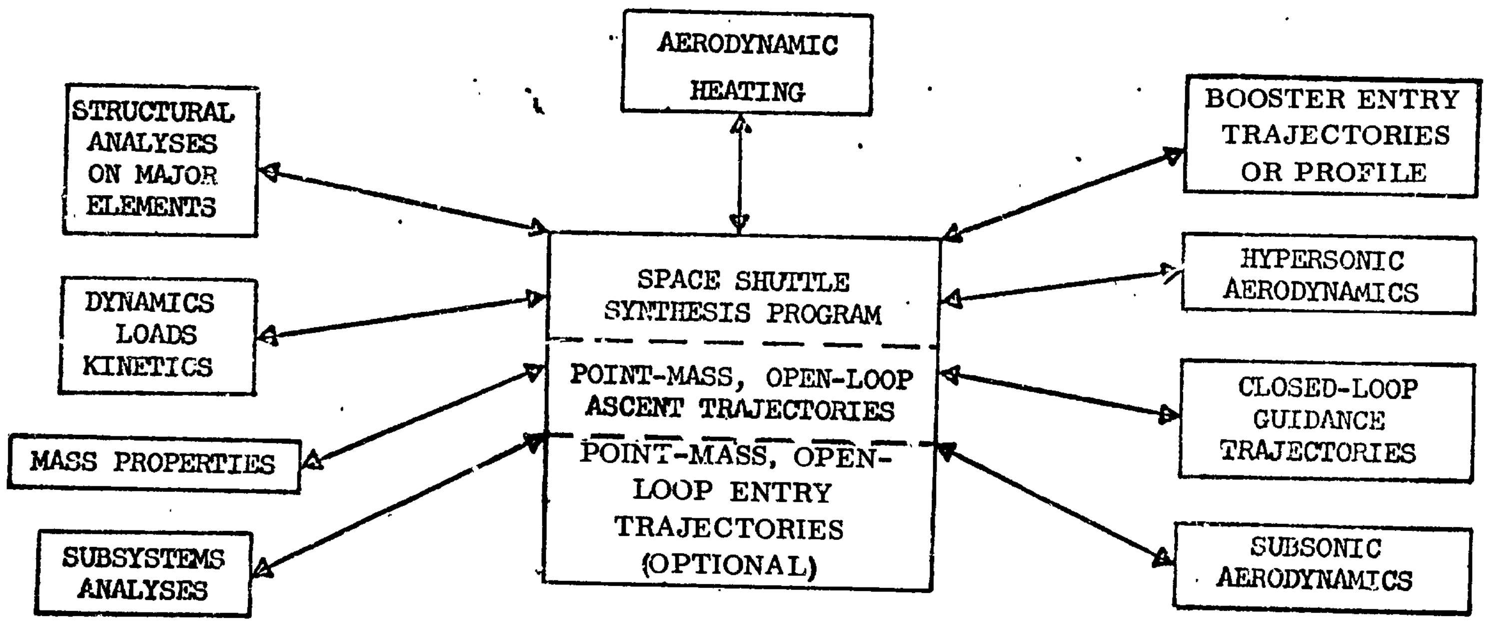


Figure 2-2. Synthesis Design Process

(2.3) will discuss the overall program organization and computational flow.

2.2.1 Weights and Geometry. The weight/volume portion of the SSSP (abbreviated WTVOL in this report) is a library of weight and volume equations for the components of space shuttle vehicles. The subprogram accepts inputs in the form of coefficients to various weights and volume equations written in terms of the geometry of a particular vehicle type. It uses existing weight data, plus inputs describing the thermal protection system, propulsion and other subsystems , as well as performance mass ratios and other mission requirements derived from the trajectory subprogram. The second generation weight breakdown in MIL-M-38310 was used as a guide to determine the level of detail and order of weight output listings described in Section 5.0. Weight equations for each component or group of components were written by incorporating appropriate provisions for varying weights correctly as the vehicle weight and/or size changes. Volume equations for important volume components are also included. An iteration process is employed so that component weights/volumes and overall weights/volumes are mutually consistent. Figure 2-3 shows the functional flow of the WTVOL subprogram.

The WTVOL subprogram solves the following basic problem: for a specified payload weight and mass ratio, find the stage gross weight and volume. This problem is solved spearately for the orbiter and booster stages, then iterations are performed to satisfy specified mission constraints or specified relationships between the booster and orbiter.

The weight equations used in WTVOL rely heavily on a unit weight approach, with any sophistication based more on selection of proper weight coefficients for input rather than on the equations themselves. This method gives the user more latitude for judgement, and permits the same equations to be used for a wide range of vehicles. To do this, however, a data library of vehicle weight coefficients

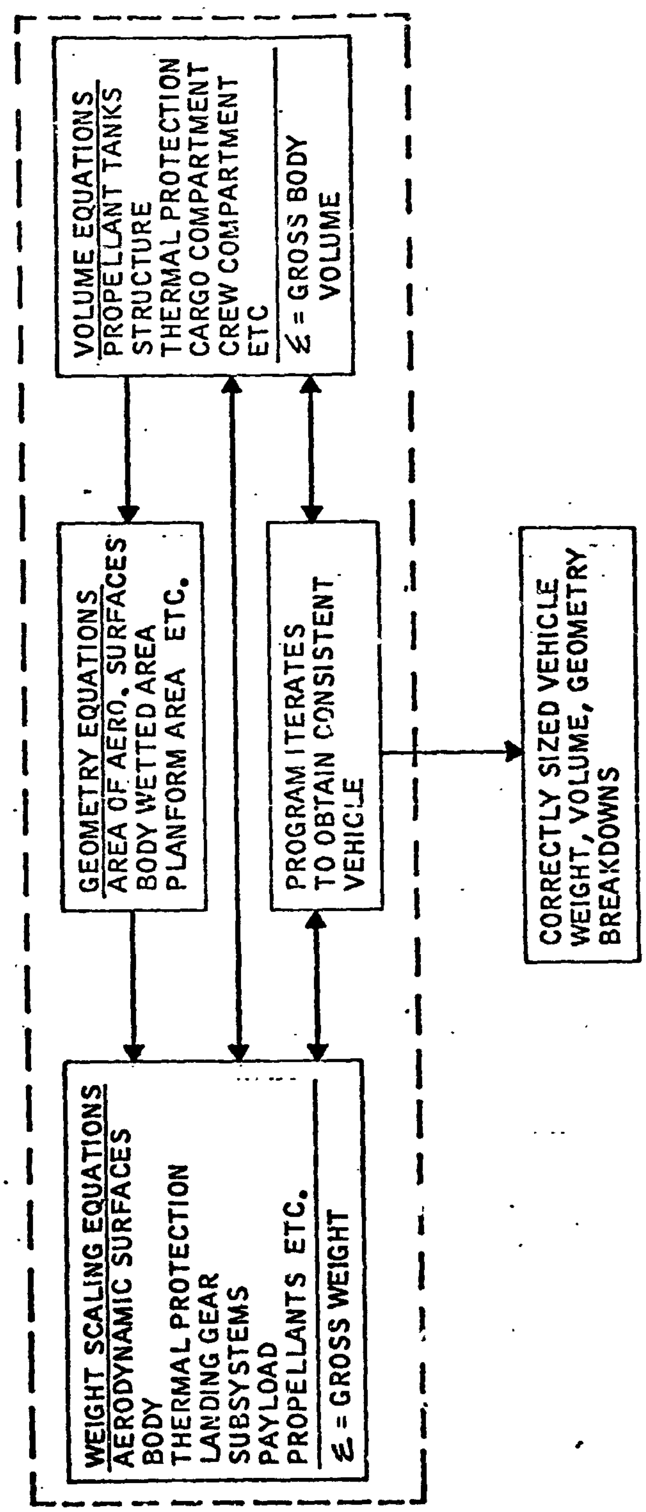


Figure 2-3. Weight/Volume Calculations

obtained from detailed design studies must be available. This is one of the objectives of the Weight/Volume Handbook, Volume II of this report. The Weight/Volume Handbook contains the compilation of all of the weight/sizing equations utilized in the WTVOL subprogram (specifically the WTSCH subroutine) and a procedure for obtaining the proper coefficients that are input for each equation. These inputs, however, are not intended to be absolute but a guide to the magnitude of the input to ensure answers of the right magnitude for the item being considered. The ideal input will always be obtained when a study of the specific design conditions for the item being considered is made, and the results of this study put in terms of the program equation input. Figure 2-4 illustrates the development of the input coefficients for the WTVOL subprogram.

2.2.2 Trajectory Computation. Introduction. The numerically integrated trajectory is computed by using a trajectory simulation module which is a special purpose version of a standard trajectory simulation program currently in use at the General Dynamics Convair Aerospace Division. This program, the General Trajectory Simulation Module (GTSM) is described in detail in Reference 1. Pertinent extracts from this document form the basis of this section and Sections 3.4 Trajectory Simulation Overlay, 4.2.3 $DATA1 (Input Parameters), and 5.1.2 Trajectory Simulation (Output Parameters). Although the ascent profile is fixed for the Space Shuttle Synthesis Program the integrated booster return trajectory is modeled by the same general methods of GTSM, consequently the program description describes the general modeling options with notes indicating those options which are specified for the ascent trajectory.

The General Trajectory Simulation Module (GTSM) program is a general purpose high speed, precision flight program which simulates the flight for an aerospace vehicle in the gravitational field of a central body. It utilizes the efficient

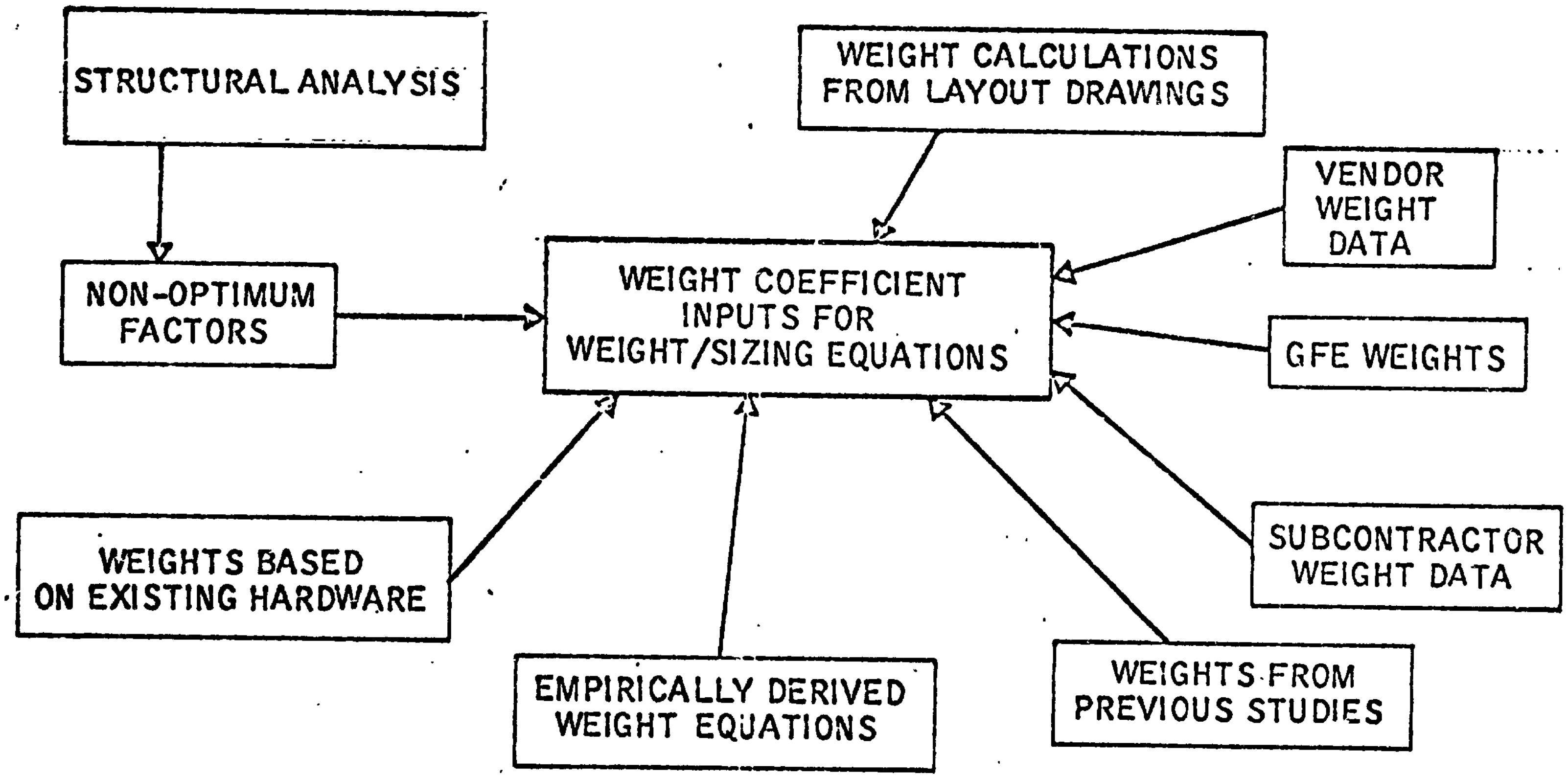


Figure 2-4. Weight/Sizing Input Development

Kutta-Merson variable stepsize numerical integration technique to integrate with respect to time the twelve state equations. These equations define the time rate of change of the three degree of freedom vehicle motion, the vehicle mass, the ideal velocity and velocity losses, and a heating parameter. The vehicle motion equations consist of three kinematic and three kinetic equations and are expressed in a natural applied force coordinate system which minimizes the extent of matrix coordinate transformations common to other simulations. The mass equation is actually a numerical model which is usually defined as part of the propulsion and fuel flow model. The ideal velocity and velocity losses equations are the propulsive, gravity, drag, and thrust misalignment acceleration equations while the heating parameter equation is the product of the relative velocity and the dynamic pressure. Numerous trajectory parameters are computed and output at each integration step. This program is characterized by its high speed, accuracy, flexible flight control options, detailed simulation model capability, and its generality of application.

Important GTSM features are:

a. Flight profiles are simulated with sectional modeling. A general staging procedure is used to terminate each simulation section upon achieving a specified value for any parameter which is computed during each integration step. An auxiliary or backup time termination is also available.

b. A powerful General Iteration Scheme (GIS) is available which eliminates unnecessary parametric computations when iterating to desired end conditions. For each of four possible GIS blocks, one or two primary control variables may be selected from any of the simulation section initializing parameters and the corresponding required end condition parameters can be any parameter which is computed at each integration step. Up to four additional parameters can be "chained" with each of the primary control variables.

c. A three-dimensional aerodynamic model is incorporated in GTSM which provides for three* common orthogonal components of the aerodynamic force. Tables for each of the corresponding aerodynamic force coefficients (yaw-normal, axial, and pitch-normal) are provided as functions of Mach number and the pitch angle of attack.

d. It is possible to maintain a specified time rate of change of the flight path angle by modulation of either the pitch angle of attack, or the bank angle. This option is particularly useful when aerodynamically controlling return trajectories.

e. Both oblate earth gravitational potential and surface geometry are available. The potential model is symmetric about the rotational axis and includes the [illegible]th harmonic term. The surface is an ellipsoid with symmetry about the rotational axis and the equatorial plane.

The GTSM program has been used very successfully to simulate trajectories for many types of vehicles and missions. Among these are:

a. SLV-3X/Agena D to synchronous equatorial orbit
b. SLV-3A/Burner II to targeted hyperbolic departure orbit
c. Atlas F targeted to impact in specified test areas
d. One and one-half Stage Reuseable Launch Vehicle to low earth orbit
e. Triamese Reuseable Launch Vehicle to low earth orbit
f. Maneuverable Lifting Entry Vehicles from parking orbit, through propulsive and aerodynamic maneuvers, to landing site area or cruise conditions
g. OV1 re-entry trajectories to impact

*In the Synthesis Program special version, the yaw-normal aerodynamic force model has been suppressed. The corresponding equations have been left intact in the program in order to minimize their incorporation should this model become required.

h. Saturn 1B to low earth orbit

i. Advanced ICBM's to targeted impact conditions, FOBS, and global ballistic trajectories

j. Air-Launched Rocket/Air Breathing Missiles

k. Low-thrust vehicle in near-earth spiral trajectories

l. Miscellaneous orbital maneuvers, coasts, and transfers including polar orbits, and multi-impulse maneuvers

The use of GTSM for these studies has significantly reduced computer expenses while providing data sufficiently accurate for predesign and several near-hardware applications. GTSM is basically as accurate as the vehicle modeling options are representative of the vehicle; the equations of motion and the integration scheme are precise and result in excellent agreement with other precision programs when the vehicle models are nearly equivalent.

Engineering Discussion. To obtain adequate simulation of an aerospace vehicle trajectory it is necessary to define the vehicle equations of motion, to evaluate the terms and coefficients of these equations, and then to solve the equations to obtain the time history of the pertinent trajectory parameters. The discussion of this section presents the three degree-of-freedom differential equations which describe the flight of an aerospace vehicle in a gravitational field of a central body with applied external forces; the required coordinate systems, transformations, and particular solutions such as crossing the poles, vertical motion, and constant time rate of change of relative flight path angle are discussed; and the vehicle modeling options which define terms in the kinetic equations of motion and provide the trajectory output.

Coordinate Systems Transformations and Equations of Motion. It is assumed that the motion of the vehicle is sufficiently described by three-degree-of-freedom kinematic and kinetic equations of motion for this analysis, and correspondingly that the rotational dynamics can be neglected and that the vehicle has the capability to achieve and maintain any specified or required attitude. There are two sets of three degree-of-freedom equations of motion widely used at Convair, the inertial system and the so called "relative" system described in Ref.1. Both systems accurately describe the motion of an aerospace vehicle in the gravitational field of a central body and subjected to aerodynamic and propulsive forces. Experience with previous analyses of lifting vehicles indicates that the relative system of Ref. 1 is more suitable for these vehicles, resulting in smoother differential flight equations, a corresponding increase in accuracy, and shorter computer run times. For these reasons this "relative" system of equations was selected. It is emphasized that this "relative" system of equations describes the same motion with respect to inertial space as the inertial system of equations does; the principle difference is that the "relative" system of equations is in terms of central body relative parameters which are expressed in a natural applied force coordinate system. The complete derivation of these equations together with all pertinent definitions are described fully in Ref. 1. The important definitions and equations are presented subsequently.

The inertial kinetic vector equation of motion is:

$$\boxed{\bar{F}_{APPLIED} + \bar{F}_g = m\,[\ddot{\bar{R}} + 2\bar{\omega} \times \dot{\bar{R}} + \bar{\omega} \times (\bar{\omega} \times \bar{R})]} \tag{2-1}$$

where:

$\bar{F}_{APPLIED}$	is the total applied force (aerodynamic and propulsive forces) vector
$\bar{F}_g$	is the force vector acting on the vehicle which is due to the gravitational field
m	is the vehicle mass
$\bar{R}$	is the current position geocentric radius vector
$\dot{\bar{R}}$	is the vector time derivative of R taken in an earth-relative coordinate system

$\ddot{\bar{R}}$ is the second vector time derivative of R taken in an earth relative coordinate system

$\bar{\omega}$ is the rotation rate vector of the earth

The relative kinematic vector equation of motion is:

$$\boxed{\left[\frac{d\bar{R}}{dt}\right] = \dot{\bar{R}} = \bar{V}} \tag{2-2}$$

where $\bar{V}$ is the earth relative velocity.
t is the time

Efficient solution of the vector equations of motion (Eq. 2-1 and 2-2) requires expression of these equations in an appropriate coordinate system. Appropriate coordinate systems and corresponding required transformations between these coordinate systems must be defined.

2.2.2.1 Coordinate Systems. It is desirable to select "natural" coordinate systems in which to express the equations of motion (Eq. 2-1 and 2-2) so that any repetitive coordinate transformation requirement is minimal. There are five coordinate systems which are used in the derivation of the GTSM scalar equations of motion.

These are the primary inertial frame, the earth-fixed frame, the local frame, the velocity frame, and the vehicle coordinate system; their definitions follow.

Primary Inertial Coordinate System. The Primary Inertial Coordinate System ($\hat{I}_1 - \hat{I}_2 - \hat{I}_3$ axes), illustrated in Figure 2-5, has its origin at the center of the central body. The first axis ($\hat{I}_1$) lies in the equatorial plane at its intersection with meridian which is defined by the longitude at the initial time. The third axis ($\hat{I}_3$) is colinear with the rotational axis of the central body (perpendicular to the equatorial plane) with positive sense toward the North Pole. The second axis ($\hat{I}_2$) forms a right-handed orthogonal coordinate system with $\hat{I}_1$ and $\hat{I}_3$ and consequently lies in the equatorial plane 90 degrees east of $\hat{I}_1$.

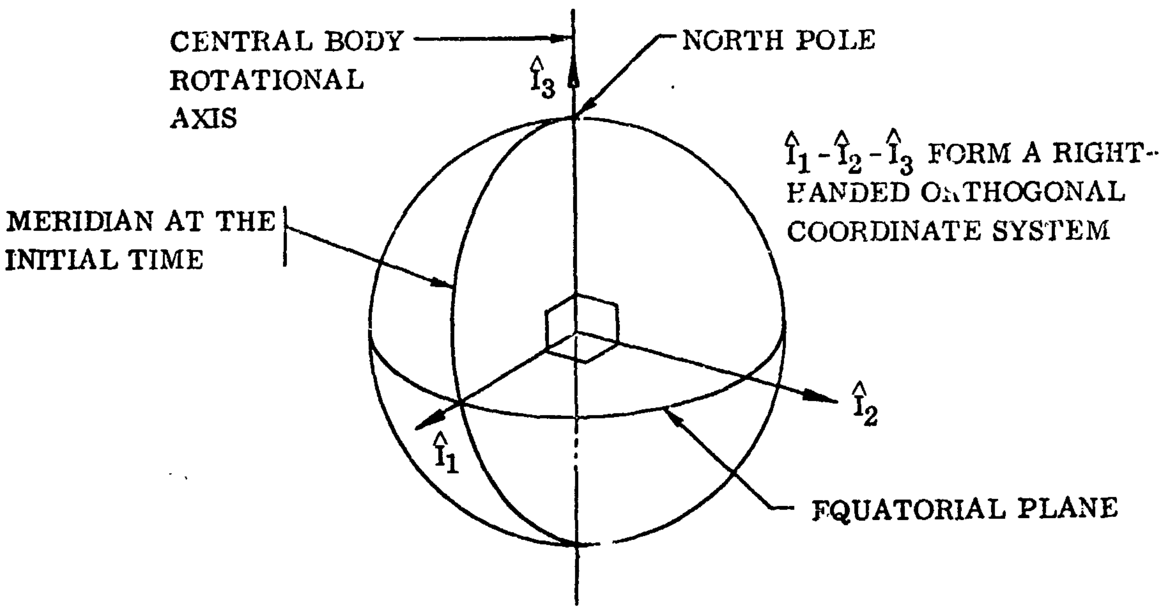


Figure 2-5. The Primary Inertial Coordinate System

The orientation of the $\hat{I}_1$-$\hat{I}_2$-$\hat{I}_3$ axes remains fixed with respect to the stars in our galaxy.

Earth-Fixed Coordinate System. The Earth-Fixed Coordinate System ($\hat{e}_1$ - $\hat{e}_2$ - $\hat{e}_3$ axes), shown in Figure 2-6, remains fixed with respect to the rotating central body. The origin of the $\hat{e}_1$ - $\hat{e}_2$ - $\hat{e}_3$ system is at the center of the central body. The first axis ($\hat{e}_1$ lies in the central body equatorial plane at a longitude of zero (Greenwich, or prime, meridian for the Earth); the second axis ($\hat{e}_2$) lies in the equatorial plane at a longitude of +90 degrees (90 degrees east of $\hat{e}_1$); and the third axis ($\hat{e}_3$) is colinear with the central body rotational axis with positive sense toward the North Pole. The $\hat{e}_1$ - $\hat{e}_2$ - $\hat{e}_3$ axes form a right-handed orthogonal coordinate system.

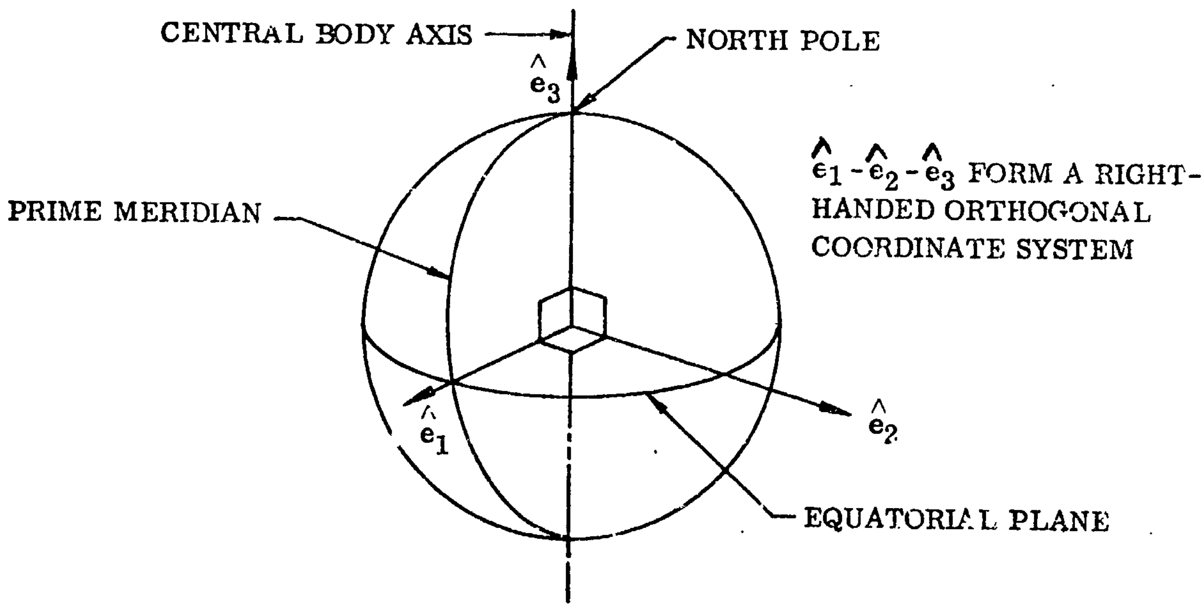


Figure 2-6. The Earth-Fixed Coordinate System

<u>Local Coordinate System.</u> The Local Coordinate System ($\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ axes), illustrated in Figure 2-7 has its origin at the current position of the aerospace vehicle point mass. The first axis ($\hat{i}_\phi$) is parallel to both the central body's equatorial plane and the local horizontal plane and points in the direction of algebraically increasing longitude (due east). The second axis ($\hat{i}_\theta$) is parallel to the tangent of the local meridian which is defined by the current longitude; the $\hat{i}_\theta$ axis points due north. The third axis ($\hat{i}_R$) is colinear with the current geocentric radius vector ($\bar{R}$) and points upward. This system of axes ($\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ coordinate system) forms a right-handed orthogonal coordinate system. It is noted that the position radius vector ($\bar{R}$), its magnitude (R), and the corresponding unit vector with the same direction ($\hat{R}$) are

identically related:

$$\bar{R} = (0)\,\hat{i}_\phi + (0)\,\hat{i}_\theta + (R)\,\hat{i}_R = R\,\hat{i}_R$$

$$\hat{R} = (0)\,\hat{i}_\phi + (0)\,\hat{i}_\theta + (1)\,\hat{i}_R = \hat{i}_R$$

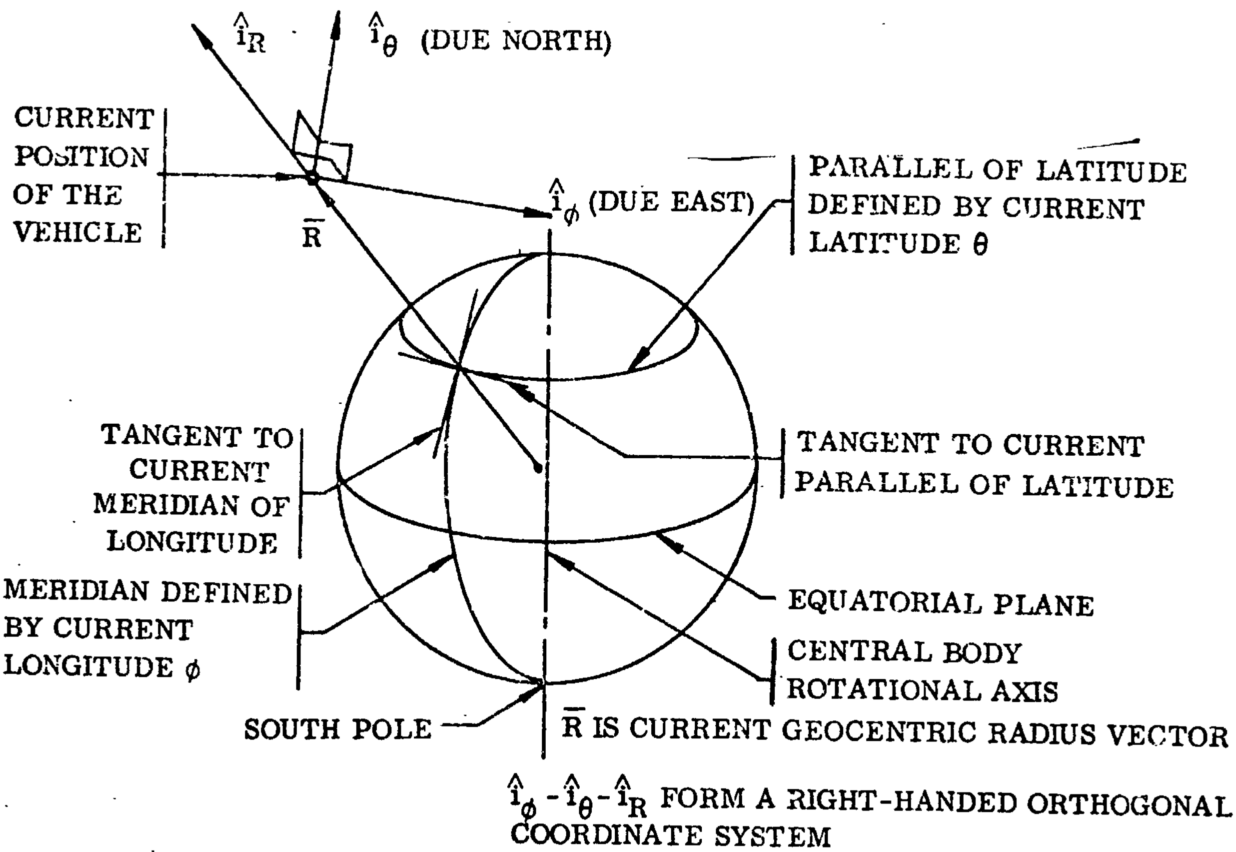


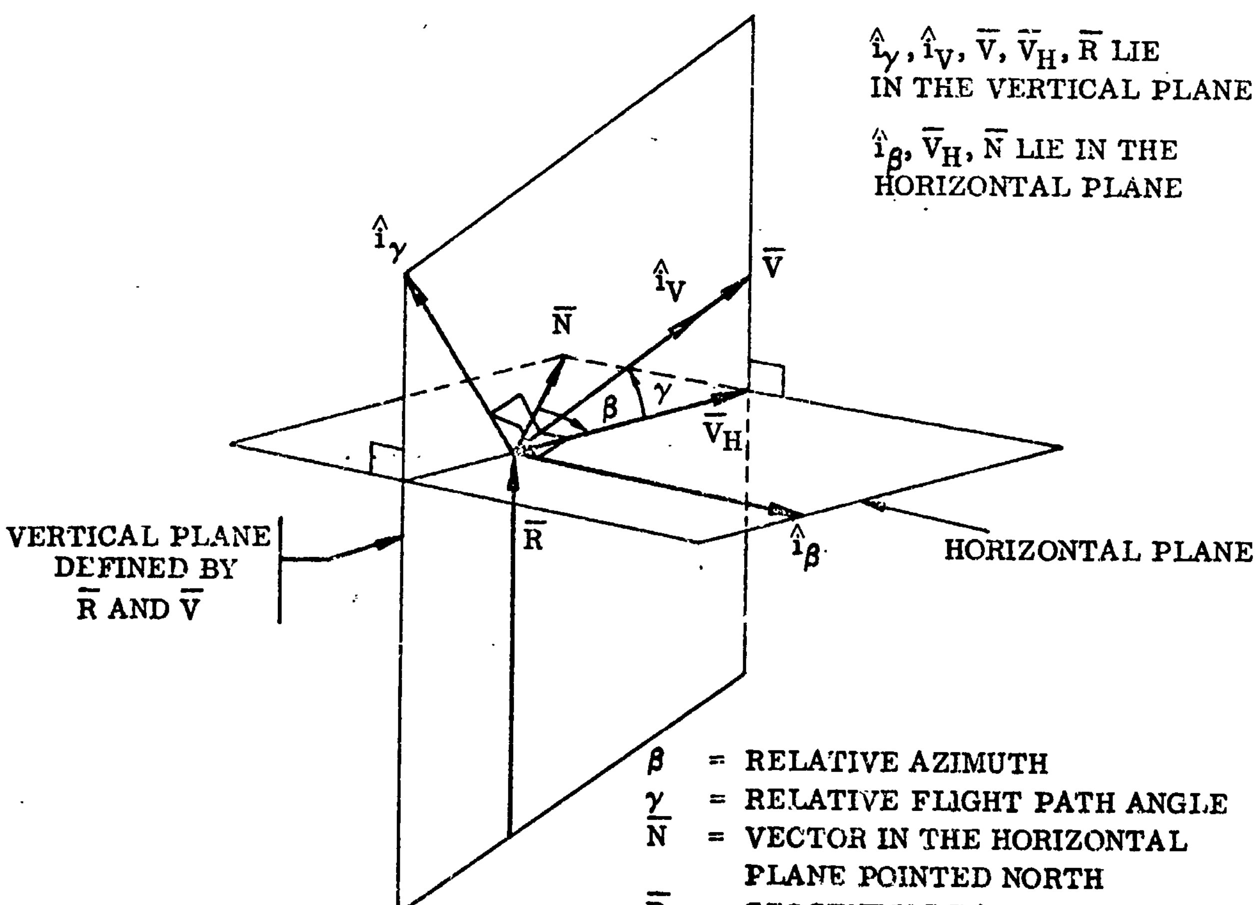
Figure 2-7. The Local Coordinate System

<u>Relative Velocity Coordinate System.</u> The Relative Velocity Coordinate System ($\hat{i}_\beta - \hat{i}_V - \hat{i}_Y$ axes) also has its origin at the current position of the aerospace vehicle point mass but has an orientation (see Figure 2-8) such that the second axis ($\hat{i}_V$) is colinear with, and has the same sense as, the relative velocity vector $\bar{V}$. The first axis ($\hat{i}_\beta$) lies in the local horizontal plane and is perpendicular to the plane defined by $\bar{V}$ and the current geocentric radius vector ($\bar{R}$). The $\hat{i}_\beta$ axis points in the direction that the relative velocity vector would move in if the relative azimuth were

algebraically increasing. The third axis ($\hat{i}_\gamma$) lies in the plane defined by $\overline{V}$ and $\overline{R}$ and is perpendicular to both $\hat{i}_\beta$ and $\hat{i}_V$. The $\hat{i}_\gamma$ axis points in the direction that the relative velocity vector would move in if the relative flight path angle were increasing algebraically. The $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ axes form a right-handed orthogonal coordinate system. It is noted that the relative velocity vector ($\overline{V}$), its magnitude (V), and the corresponding unit vector with the same direction ($\hat{V}$) are identically related

$$\overline{V} = (0)\,\hat{i}_\beta + (V)\,\hat{i}_V + (0)\,\hat{i}_\gamma = V\,\hat{i}_V$$

$$\hat{V} = (0)\,\hat{i}_\beta + (1)\,\hat{i}_V + (0)\,\hat{i}_\gamma = \hat{i}_V$$



β = RELATIVE AZIMUTH
γ = RELATIVE FLIGHT PATH ANGLE
$\overline{N}$ = VECTOR IN THE HORIZONTAL PLANE POINTED NORTH
$\overline{R}$ = GEOCENTRIC RADIUS VECTOR
$\overline{V}$ = RELATIVE VELOCITY VECTOR
$\overline{V}_H$ = RELATIVE VELOCITY VECTOR PROJECTION IN THE HORIZONTAL PLANE

$\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ FORM A RIGHT HANDED ORTHOGONAL COORDINATE SYSTEM

Figure 2-8 The Relative Velocity Coordinate System

Vehicle Coordinate System. The Vehicle Coordinate System ($\hat{\xi}$ - $\hat{\eta}$ - $\hat{\zeta}$ axes) which is illustrated in Figure 2-9 has its origin located on the longitudinal center axis of the vehicle (usually the longitudinal axis of symmetry) at the aft vehicle section. The first axis ($\hat{\xi}$ axis) is the roll axis and is colinear with the longitudinal center axis and is pointed forward. The second axis (the $\hat{\eta}$ axis) is the pitch axis and is positive to the right of the vehicle. The third axis (the $\hat{\zeta}$ axis) is the yaw axis and is positive toward the bottom of the vehicle. The $\hat{\xi}$ - $\hat{\eta}$ - $\hat{\zeta}$ axes form an orthogonal right-handed coordinate system which is in accordance with the coordinate system specifications listed in References 2, 3, and 4.

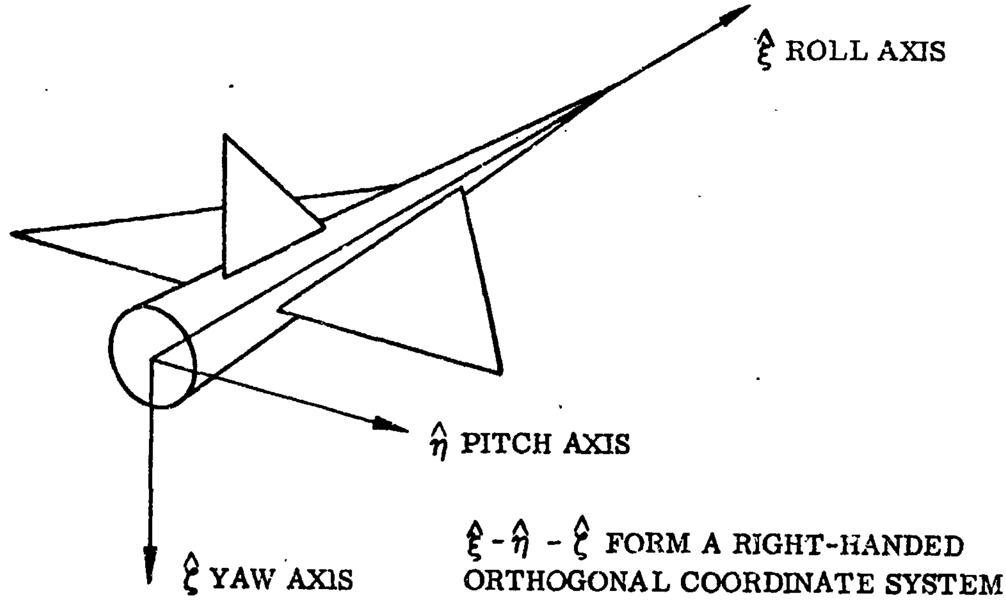


Figure 2-9. The Vehicle Coordinate System

2.2.2.2 Coordinate System Transformations. To derive the "natural" applied force equations of motion used in GTSM, one must express all elements of each quation of motion (Eq. 2-1 and 2-2) in the same coordinate system. Consequently it is first necessary to specify which coordinate system is to be used for a particular equation and then identify and define the required coordinate transformations. It is desirable to express Eq. 2-1 in the Relative Velocity Coordinate System ($\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ axes) and Eq. 2-2 in the Local Coordinate System ($\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ axes). Since the expanded forms of these equations contain elements which are readily expressed in other coordinate systems, it is necessary to define the required transformations to both the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ and $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ systems. This is accomplished by repeated application of the elemental coordinate transformation matrices which are presented in Appendix I.

Transformation from Vehicle Axes to Relative Velocity Axes. Determination of the aerodynamic forces acting on the vehicle requires knowing the pitch and yaw orientation with respect to the relative velocity. Expression of the vehicle propulsive forces in the relative velocity frame requires knowing the vehicle orientation with respect to the relative velocity. The transformation matrix for the Vehicle Axes to Relative Velocity Axes transformation is obtained by assuming an initial alignment of these two coordinate systems (see Figure 2-10a) such that the $\hat{\xi}$ axis is colinear with the $\hat{i}_\beta$ axis with the same sense, the $\hat{\eta}$ axis is colinear with the $\hat{i}_V$ axis with the same sense, and the $\hat{\zeta}$ axis is colinear with the $\hat{i}_\gamma$ axis with the same sense.

The inverse transformation from the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system to the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ system is then defined by using four sequenced rotations of the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ axes about the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ axes consisting of:

a. The first two rotations are comprised of; first a rotation about the yaw axis ($\hat{\zeta}$ axis) through +90 degrees, and then secondly about the new roll axis ($\hat{\xi}$ axis) through 180 degrees. The algebraic sign is in accordance with right-handed rotations of the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ axes. If A is any vector expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system prior to these two rotations, then these rotations are expressed in matrix form as:

$$\begin{pmatrix} A_v \\ A_\beta \\ -A_\gamma \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} A_\beta \\ A_v \\ A_\gamma \end{pmatrix} \qquad (2\text{-}3)$$

The resulting alignment of the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ axes is shown in Figure 2-10b.

b. The third rotation is about the roll ($\hat{\xi}$ axis) through the roll (bank) angle (σ). The algebraic sign is in accordance with right-handed rotations of the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ axes.

c. The fourth rotation is about the pitch axis ($\hat{\eta}$ axis) through the pitch angle of attack (α). The algebraic sign is in accordance with right-handed rotations of the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ axes.

d. The final rotation is about the yaw axis ($\hat{\zeta}$ axis) through the yaw angle of attack(λ). The algebraic sign is in accordance with right-handed rotations of the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ axes.

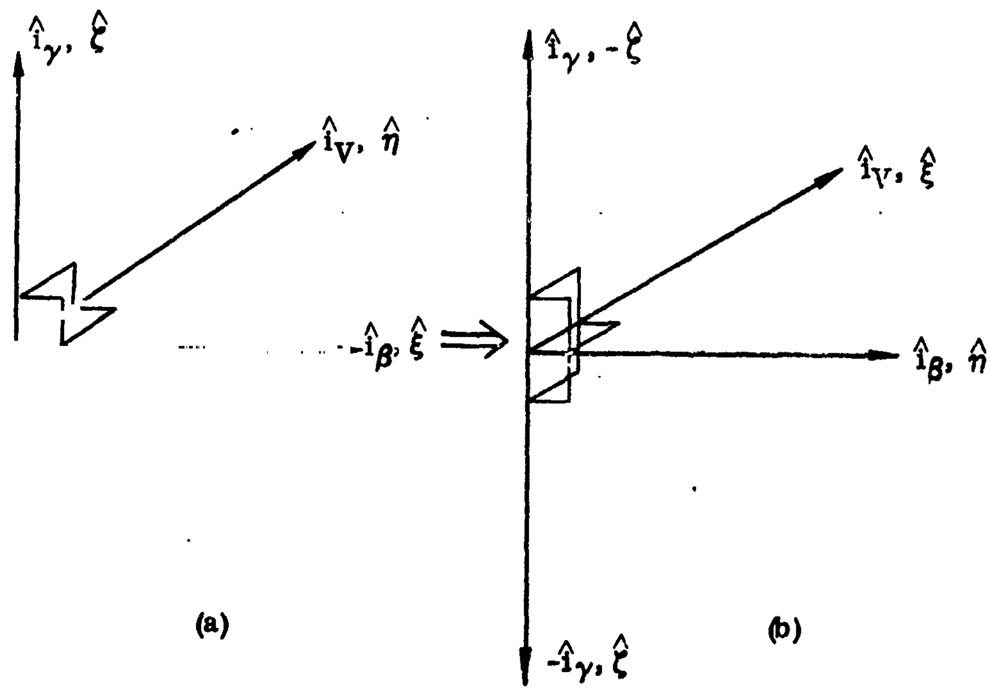


Figure 2-10. Axes Alignment Initially and During the Vehicle Axes to Relative Velocity Axes Transformation

If $\bar{A}$ is any vector which is expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system, then successive applications of the elemental coordinate transformation matrices (A-M1), (A-M2), and (A-M3) in that order for σ, α, and λ rotation angles, respectively, yields A expressed in the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ system.

$$\begin{pmatrix} A_\xi \\ A_\eta \\ A_\zeta \end{pmatrix} = \begin{pmatrix} \cos\lambda & \sin\lambda & 0 \\ -\sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\sigma & \sin\sigma \\ 0 & -\sin\sigma & \cos\sigma \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} A_\beta \\ A_V \\ -A_\gamma \end{pmatrix} \qquad (2\text{-}4)$$

where $\bar{A} = A_\xi \hat{\xi} + A_\eta \hat{\eta} + A_\zeta \hat{\zeta} = A_V \hat{j}_V + A_\beta \hat{j}_\beta + (-A_\gamma) \hat{j}_\gamma$

By carrying out the matrix multiplications indicated in Eq. 2-4, and noting that for orthogonal coordinate system transformations the inverse matrix is equal to the transpose matrix, Eq. 2-4 becomes:

$$\begin{pmatrix} A_\beta \\ A_V \\ A_\gamma \end{pmatrix} = \begin{pmatrix} (+\cos\lambda\sin\alpha\sin\sigma + \sin\lambda\cos\sigma) & (-\sin\lambda\sin\alpha\sin\sigma + \cos\lambda\cos\sigma) & (-\cos\alpha\sin\sigma) \\ (+\cos\lambda\cos\alpha) & (-\sin\lambda\cos\alpha) & (+\sin\alpha) \\ (+\cos\lambda\sin\alpha\cos\sigma - \sin\lambda\sin\sigma) & (-\sin\lambda\sin\alpha\cos\sigma - \cos\lambda\sin\sigma) & (-\cos\alpha\cos\sigma) \end{pmatrix} \begin{pmatrix} A_\xi \\ A_\eta \\ A_\zeta \end{pmatrix} \qquad (M\text{-}1)$$

<u>Transformation from Local Axes to Relative Velocity Axes.</u> The transformation matrix for the Local Axes to Relative Velocity Axes transformation is readily obtained by assuming an initial alignment of the Velocity Axes with the Local Axes such that the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ axes are respectively parallel to and have the same sense as the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ axes. It is necessary to define two parameters; γ is the earth relative flight path angle measured positively upward from the local horizontal, and β is the earth relative azimuth measured clockwise from north. Two sequenced rotations of the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system about the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ system after the initial alignment define the transformation matrix. The sequenced rotations are:

a. A first rotation about the $\hat{i}_\gamma$ axis through the negative of the relative azimuth ($-\beta$).

b. A second rotation about the $\hat{i}_\beta$ axis through the current relative flight path angle (γ).

The algebraic sign for both rotations is in accordance with right-handed rotations of the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ axes. If $\bar{A}$ is any vector which is expressed in the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ system, then successive applications of the fundamental coordinate transformations (A-M3) and (A-M1), in that order, for the $-\beta$ and γ rotation angles respectively, yields $\bar{A}$ expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system. It is noted that by convention the azimuth (β) is measured clockwise from North, consequently the first rotation is through a $-\beta$ angle since a $+\beta$ rotation would require a left-handed rotation about the $\hat{i}_\gamma$ axis.

$$\begin{pmatrix} A_\beta \\ A_V \\ A_\gamma \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} \cos(-\beta) & \sin(-\beta) & 0 \\ -\sin(-\beta) & \cos(-\beta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} A_\phi \\ A_\theta \\ A_R \end{pmatrix} \qquad (2\text{-}5)$$

Carrying out the indicates matrix multiplications, Equation 2-5 becomes:

$$\begin{pmatrix} A_\beta \\ A_V \\ A_\gamma \end{pmatrix} = \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta\cos\gamma & \cos\beta\cos\gamma & \sin\gamma \\ -\sin\beta\sin\gamma & -\cos\beta\sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} A_\phi \\ A_\theta \\ A_R \end{pmatrix} \qquad (M\text{-}2)$$

<u>Transformation from Earth-Fixed Axes to Local Axes.</u> The transformation matrix for Earth-Fixed Axes to Local Axes is obtained in a manner similar to that used for the previously derived transformation (Eq. M-2). In this case the Local Axes are assumed to be initially aligned with the Earth-Fixed Axes such that the

$\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ axes are parallel to and have the same sense as the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ axes, respectively. Two parameters must be defined; θ is the geocentric latitude and is positive in the northern hemisphere, and ϕ is the geocentric longitude which is measured positively to the east of the Greenwich Meridian. Two sequenced rotations of the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ system about the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ system are required after the initial alignment to define the transformation matrix. The sequenced rotations are:

a. The first rotation is about the $\hat{i}_R$ axis through an angle equal to the longitude plus 90 degrees ($\phi + 90°$). It is necessary to add 90 degrees to direct the first axis ($\hat{i}_\phi$) due east at the completion of the transformation.

b. A second rotation is the $\hat{i}_\phi$ axis through the complement of the geocentric latitude ($90° - \theta$). The complement is required to place the $\hat{i}_R$ axis in the up radial direction upon completion of the transformation.

The algebraic sign for both rotations is in accordance with right-handed rotations of the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ axes. If $\bar{A}$ is any vector which is expressed in the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ system, then successive applications of the elemental coordinate transformations (A-M3) and (A-M1), in that order, for the $\phi + 90°$ and $90° - \theta$ rotation angles, respectively, yields $\bar{A}$ expressed in the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ system.

$$\begin{pmatrix} A_\phi \\ A_\theta \\ A_R \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(90°-\theta) & \sin(90°-\theta) \\ 0 & -\sin(90°-\theta) & \cos(90°-\theta) \end{pmatrix} \begin{pmatrix} \cos(\phi+90°) & \sin(\phi+90°) & 0 \\ -\sin(\phi+90°) & \cos(\phi+90°) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Ae_1 \\ Ae_2 \\ Ae_3 \end{pmatrix} \quad (2\text{-}6)$$

Executing the matrix multiplication indicated in Eq. 2-6 leads to:

$$\begin{pmatrix} A_\phi \\ A_\theta \\ A_R \end{pmatrix} = \begin{pmatrix} -\sin\phi & \cos\phi & 0 \\ -\cos\phi \sin\theta & -\sin\phi \sin\theta & \cos\theta \\ \cos\phi \cos\theta & \sin\phi \cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} Ae_1 \\ Ae_2 \\ Ae_3 \end{pmatrix} \quad (\text{M-3})$$

<u>Transformation from Earth-Fixed Axes to Relative Velocity Axes</u>. The transformation matrix for Earth-Fixed Axes to Local Axes is readily obtained by combining the two transformations represented by Eq. M-2 and M-3 . The Relative Velocity Axes are assumed to be initially aligned with the Earth-Fixed Axes, rotated to the orientation of the Local Axes defined by ϕ and θ, and then rotated to the required orientation for the Relative Velocity Axes defined by β and γ. If $\bar{A}$ is any vector which is expressed in the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ system, then successive application of the transformations defined by Eq. M-3 and M-2 in that order yields $\bar{A}$ expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system.

$$\begin{pmatrix} A_\beta \\ A_V \\ A_\gamma \end{pmatrix} = \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta\cos\gamma & \cos\beta\cos\gamma & \sin\gamma \\ -\sin\beta\sin\gamma & -\cos\beta\sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} -\sin\phi & \cos\phi & 0 \\ -\cos\phi\sin\theta & -\sin\phi\sin\theta & \cos\theta \\ \cos\phi\cos\theta & \sin\phi\cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} Ae_1 \\ Ae_2 \\ Ae_3 \end{pmatrix} \quad (2\text{-}7)$$

Carrying out the matrix multiplication indicated in Eq. 2-7, the required transformation is obtained:

$$\begin{pmatrix} A_\beta \\ A_V \\ A_\gamma \end{pmatrix} = \begin{pmatrix} [-\cos\beta\sin\phi + \sin\beta\cos\phi\sin\theta] & [+\cos\beta\cos\phi + \sin\beta\sin\phi\sin\theta] & [-\sin\beta\cos\theta] \\ \left[\begin{matrix} -\sin\beta\cos\gamma\sin\phi - \cos\beta\cos\gamma\cos\phi\sin\theta \\ +\sin\gamma\cos\phi\cos\theta \end{matrix}\right] & \left[\begin{matrix} +\sin\beta\cos\gamma\cos\phi - \cos\beta\cos\gamma\sin\phi\sin\theta \\ +\sin\gamma\sin\phi\cos\theta \end{matrix}\right] & [+\cos\beta\cos\gamma\cos\theta + \sin\gamma\sin\theta] \\ \left[\begin{matrix} +\sin\beta\sin\gamma\sin\phi + \cos\beta\sin\gamma\cos\phi\sin\theta \\ +\cos\gamma\cos\phi\cos\theta \end{matrix}\right] & \left[\begin{matrix} -\sin\beta\sin\gamma\cos\phi + \cos\beta\sin\gamma\sin\phi\sin\theta \\ +\cos\gamma\sin\phi\cos\theta \end{matrix}\right] & [-\cos\beta\sin\gamma\cos\theta + \cos\gamma\sin\theta] \end{pmatrix} \begin{pmatrix} Ae_1 \\ Ae_2 \\ Ae_3 \end{pmatrix} \quad (M\text{-}4)$$

<u>Transformation from Relative Velocity Axes to Local Axes</u>. The transformation from the Relative Velocity Axes to the Local Axes is the inverse operation from that described by Eq. M-2, Transformation from Local Axes to Relative Velocity Axes. Since this transformation is between orthogonal coordinate systems, the inverse of the transformation matrix given by Eq. M-2 is equal to the transpose of the matrix. Consequently, if $\bar{A}$ is any vector which is expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system, application of the transpose of the transformation matrix given by Eq. M-2 yields $\bar{A}$ expressed in the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ system.

$$\begin{pmatrix} A_\phi \\ A_\theta \\ A_R \end{pmatrix} = \begin{pmatrix} \cos\beta & \sin\beta\cos\gamma & -\sin\beta\sin\gamma \\ -\sin\beta & \cos\beta\cos\gamma & -\cos\beta\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} A_\beta \\ A_V \\ A_\gamma \end{pmatrix} \qquad \text{(M-5)}$$

2.2.2.3 Three Degree-of-Freedom Scalar Differential Equations of Motion. The solution of Eq. 2-1 and 2-2 yields the time history for the pertinent trajectory parameters. To solve these equations, it is necessary to express them as scalar equations. This is accomplished by first expressing them in a convenient coordinate system and then equating the vector components along like coordinate axes. Two sets of scalar equations arise; the kinematic scalar equations and the kinetic scalar equations.

Kinematic Scalar Equations of Motion. Eq. 2-2 can be reduced to three kinematic differential equations of motion. The technique which is employed to obtain these equations is to express both $\dot{\bar{R}}$ and $\bar{V}$ in one of the defined coordinate systems, perform the most simple sequence of transformations which results in both $\dot{\bar{R}}$ and $\bar{V}$ being expressed in the same coordinate system, and then equate like vector components. The $\dot{\bar{R}}$ term is readily expressed in the Local Coordinate System, $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ while the $\bar{V}$ is first expressed in the Relative Velocity System, $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$, and then transformed into the Local Coordinate System.

Examination of the geometry of the Local Coordinate System (see Figure 2-7) indicates that the $\dot{\bar{R}}$ component along the $\hat{i}_\phi$ axis (direction of increasing longitude) is $R\,\dot{\phi}\cos\theta$, along the $\hat{i}_\theta$ axis (direction of increasing latitude) is $R\,\dot{\theta}$, and along the $\hat{i}_R$ axis (direction of increasing geocentric radius magnitude) is $\dot{R}$. Thus $\dot{\bar{R}}$ in vector form is:

$$\dot{\bar{R}} = (R\,\dot{\phi}\cos\theta)\,\hat{i}_\phi + (R\dot{\theta})\,\hat{i}_\theta + (\dot{R})\,\hat{i}_R \qquad \text{(2-8)}$$

It is easiest to express $\bar{V}$ in the Relative Velocity Coordinate System, since by definition of this system, the direction of the $\hat{i}_V$ axis is the same as $\bar{V}$, consequently $\dot{\bar{V}}$ is expressed

$$\bar{V} = (0)\,\hat{i}_\beta + V\,\hat{i}_V + (0)\,\hat{i}_\gamma = V\,\hat{i}_V \tag{2-9}$$

Both the transformations defined by Eq. M-2 and M-5 have the same terms. However, Eq. 2-9 is simpler than Eq. 2-8, consequently it appears easier to transform Eq. 2-9 to the Local Coordinate System than to transform Eq. 2-8 to the Relative Velocity System. Eq. 2-9 is easily transformed into the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ Coordinate System by application of the coordinate transformation matrix [M-5]*.

$$\begin{pmatrix} V_\phi \\ V_\theta \\ V_R \end{pmatrix} = [M\text{-}5]\begin{pmatrix} 0 \\ V \\ 0 \end{pmatrix} = (V \sin\beta\cos\gamma)\,\hat{i}_\phi + (V\cos\beta\cos\gamma)\,\hat{i}_\theta + (V\sin\gamma)\,\hat{i}_R = \bar{V} \tag{2-10}$$

Since Eq. 2-10 is expressed in the same coordinate system as Eq. 2-8, no further transformations are required. By the relationship of Eq. 2-2, $\dot{\bar{R}}$ which is given by Eq. 2-8 is equal to $\bar{V}$ which is given by Eq. 2-10; consequently in order to satisfy this equivalence, like components of each equation must be equal. Equating these components yields:

$$\dot{R} = V\sin\gamma \quad \Rightarrow \quad \boxed{\dot{R} = V\sin\gamma} \tag{2-11}$$

$$R\dot{\theta} = V\cos\beta\cos\gamma \quad \Rightarrow \quad \boxed{\dot{\theta} = \frac{V\cos\beta\cos\gamma}{R}} \tag{2-12}$$

$$R\cos\theta\,\dot{\phi} = V\sin\beta\cos\gamma \quad \Rightarrow \quad \boxed{\dot{\phi} = \frac{V\sin\beta\cos\gamma}{R\cos\theta}} \tag{2-13}$$

*[M-5] denotes the transformation matrix contained in Eq. M-5

Equations 2-11, 2-12, and 2-13 are the scalar representations of Eq. 2-2 and are consequently the three scalar kinematic equations of motion. These equations define the geometry of motion in terms of local position and relative velocity parameters (R, θ, ϕ, V, γ, β).

Kinetic Scalar Differential Equations of Motion. Eq. 2-1 can be reduced to three kinetic differential equations of motion by a technique similar to that used to obtain the three kinematic equations of motion. The principal differences are that Eq. 2-1 has more terms and consequently requires more manipulation than Eq. 2-2 and that it is desirable to transform all terms of the kinetic equation into the Relative Velocity Coordinate System rather than the Local Coordinate System. The procedure that is used consists of:

a. Each term of Eq. 2-1 is expressed in the most convenient coordinate system.
b. Each term is transformed into the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system, if it is not already expressed in that system, by the most simple sequence of transformations.
c. The vector components along like coordinate axes, which must be identically equal to satisfy Eq. 2-1, are equated to obtain the three kinetic scalar equations of motion.

The applied force vector ($\bar{F}_{APPLIED}$) of Eq. 2-1 which consists of both Aerodynamic and propulsive forces is conveniently expressed in the Vehicle Axes (see Figure 2-9). The three components of $\bar{F}_{APPLIED}$ are defined:

F_ξ = Axial thrust plus the aerodynamic* axial force. These forces act along the $\hat{\xi}$ axis.

F_η = Aerodynamic yaw force which acts along the $\hat{\eta}$ axis.

F_ζ = Aerodynamic normal force which acts along the $\hat{\zeta}$ axis.

* Algebraic sense

The algebraic sign for F_ξ, F_η, and F_ζ is in accordance with the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ coordinate system. In vector form, $\bar{F}_{APPLIED}$ is given by

$$\bar{F}_{APPLIED} = F_\xi \hat{\xi} + F_\eta \hat{\eta} + F_\zeta \hat{\zeta} \tag{2-14}$$

The $\bar{F}_{APPLIED}$ vector can be expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system by application of the coordinate transformation matrix $[M-1]^\dagger$

$$\begin{pmatrix} F_\beta \\ F_V \\ F_\gamma \end{pmatrix} = [M-1] \begin{pmatrix} F_\xi \\ F_\eta \\ F_\zeta \end{pmatrix} \quad \Longrightarrow \quad \bar{F}_{APPLIED} = F_\beta \hat{i}_\beta + F_V \hat{i}_V + F_\gamma \hat{i}_\gamma$$

where:

$$\left.\begin{aligned} F_\beta &= (+\cos\lambda \sin\alpha \sin\sigma + \sin\lambda \cos\sigma)F_\xi + (-\sin\lambda \sin\alpha \sin\sigma + \cos\lambda \cos\sigma)F_\eta + (-\cos\alpha \sin\sigma)F_\zeta \\ F_V &= (+\cos\lambda \cos\alpha)F_\xi + (-\sin\lambda \cos\alpha)F_\eta + (+\sin\alpha)F_\zeta \\ F_\gamma &= (+\cos\lambda \sin\alpha \cos\sigma - \sin\lambda \sin\sigma)F_\xi + (-\sin\lambda \sin\alpha \cos\sigma - \cos\lambda \sin\sigma)F_\eta + (-\cos\alpha \cos\sigma)F_\zeta \end{aligned}\right\} \tag{2-15}$$

The gravitational force vector ($\bar{F}_g$) of Eq. 2-1 is easily expressed in the Local Coordinate System (see Figure 2-7)

$$\bar{F}_g = m\,[(0)\,\hat{i}_\phi + g_\theta \hat{i}_\theta - g_R \hat{i}_R] = m\,[g_\theta \hat{i}_\theta - g_R \hat{i}_R] \tag{2-16}$$

Application of the $[M-2]^*$ coordinate transformation matrix to Eq. 2-16 yields the $\bar{F}_g$ vector expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system

$$\begin{pmatrix} F_{g_\beta} \\ F_{g_V} \\ F_{g_\gamma} \end{pmatrix} = [M-2](m) \begin{pmatrix} 0 \\ g_\theta \\ -g_R \end{pmatrix} \quad \Longrightarrow \quad \bar{F}_g = F_{g_\beta} \hat{i}_\beta + F_{g_V} \hat{i}_V + F_{g_\gamma} \hat{i}_\gamma \tag{2-17}$$

† [M-1] denotes the transformation matrix contained in Eq. (M-1).

* [M-2] denotes the transformation matrix contained in Eq. (M-2)

$$\bar{F}_g = m\,[(-g_\theta \sin\beta)\hat{i}_\beta + (g_\theta \cos\beta \cos\gamma - g_R \sin\gamma)\hat{i}_V + (-g_\theta \cos\beta \sin\gamma - g_R \cos\gamma)\hat{i}_\gamma]$$

where: g_θ is the gravitational acceleration along the $\hat{i}_\theta$ axis and g_R is the gravitational acceleration along the $-\hat{i}_R$ axis.

The earth-relative acceleration R is obtained from its definition:

$$\ddot{\bar{R}} = \left(\frac{d\bar{V}}{dt}\right) = \dot{\bar{V}} \tag{2-18}$$

$\dot{\bar{V}}$ can be considered as being composed of motion in a coordinate system which is moving with respect to the Earth-Fixed Coordinate System. If this moving system is taken to be the Relative Velocity Coordinate System, then by noting that

$$\dot{\bar{A}} = \dot{\bar{A}} + \bar{\omega} \times \bar{A} \tag{2-19}$$

where $\bar{A}$ an arbitrary time varying vector expressed in both the non-rotating reference frame and the rotating frame

$\dot{\bar{A}}$ is the vector time derivative of ^ taken in the non-rotating reference frame

$\dot{\bar{A}}$ is the vector time derivative of A taken in the rotating frame

$\bar{\omega}$ is the rotation rate vector of the rotating frame about the non-rotating reference frame

Eq. 2-18 can be written:

$$\ddot{\bar{R}} = \left(\frac{d\bar{V}}{dt}\right) = \left(\frac{d\bar{V}}{dt}\right)_{/RVC} + \bar{\omega}_{V/F} \times \bar{V} \tag{2-20}$$

where: the subscript RVC denotes that the derivative is taken in the Relative Velocity Coordinate System.

$\hat{\omega}_{V/F}$ is the rotational rate and direction vector of the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system about the $\hat{i}_{e_1} - \hat{i}_{e_2} - \hat{i}_{e_3}$ system.

The rotation of the Local System about the Earth-Fixed System (defined by the rotational rate and direction vector $\bar{\omega}_{L/F}$) together with the rotation of the Relative Velocity System about the Local System (defined by the rotational rate and direction vector $\bar{\omega}_{V/L}$) comprise the rotation of the Relative Velocity System about the Earth-Fixed System (expressed by $\bar{\omega}_{V/F}$). In equation form, this composition is expressed:

$$\bar{\omega}_{V/F} = \bar{\omega}_{L/F} + \bar{\omega}_{V/L} \tag{2-21}$$

From the geometry of the coordinate systems (see Figures 2-6, 2-7, and 2-8) the $\bar{\omega}_{L/F}$ and $\bar{\omega}_{V/L}$ vectors can be expressed as:

$$\left.\begin{aligned} \bar{\omega}_{L/F} &= (-\dot{\theta})\hat{i}_\phi + (+\dot{\phi})\hat{e}_3 \\ \bar{\omega}_{V/L} &= (+\dot{\gamma})\hat{i}_\beta + (-\dot{\beta})\hat{i}_R \end{aligned}\right\} \tag{2-22}$$

Equations (2-22) when combined according to Eq. (2-21) yields

$$\bar{\omega}_{V/F} = -\dot{\theta}\,\hat{i}_\phi + \dot{\phi}\,\hat{e}_3 + \dot{\gamma}\,\hat{i}_\beta - \dot{\beta}\,\hat{i}_R \tag{2-23}$$

Equations 2-22 are obviously "mixed mode", that is, their terms are expressed in various coordinate systems. This is permissible since in each equation, the two rotation components are perpendicular and consequently are mathematically independent. Before using Eq. 2-23 to expand Eq. 2-20 it is necessary to express all terms of Eq. 2-23 in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system. This is accomplished with the transformation matrices [M-2] * and[M-4] * . Let $X = \dot{\phi}\,\hat{e}_3$, $\overline{Y} = -\dot{\theta}\, i_\phi - \dot{\beta}\,\hat{i}_R$, and $\overline{Z} = \dot{\gamma}\,\hat{i}_\beta$, then:

$$\begin{pmatrix} X_\beta \\ X_V \\ X_\gamma \end{pmatrix} = [M\text{-}4]\begin{pmatrix} 0 \\ 0 \\ \dot{\phi} \end{pmatrix} \quad \Longrightarrow \quad \overline{X} = X_\beta \hat{i}_\beta + X_V \hat{i}_V + X_\gamma \hat{i}_\gamma$$

where:

$$\left.\begin{aligned} X_\beta &= -\dot{\phi}(\sin\beta\cos\theta) \\ X_V &= \dot{\phi}(\cos\beta\cos\gamma\cos\theta + \sin\gamma\sin\theta) \\ X_\gamma &= \dot{\phi}(-\cos\beta\sin\gamma\cos\theta + \cos\gamma\sin\theta) \end{aligned}\right\} \qquad (2\text{-}24)$$

and

$$\begin{pmatrix} Y_\beta \\ Y_V \\ Y_\gamma \end{pmatrix} = [M\text{-}2]\begin{pmatrix} -\dot{\theta} \\ 0 \\ -\dot{\beta} \end{pmatrix}. \quad \Longrightarrow \quad \overline{Y} = Y_\beta \hat{i}_\beta + Y_V \hat{i}_V + Y_\gamma \hat{i}_\gamma$$

where

$$\left.\begin{aligned} Y_\beta &= -\dot{\theta}(\cos\beta) \\ Y_V &= -\dot{\theta}(\sin\beta\cos\gamma) - \dot{\beta}(\sin\gamma) \\ Y_\gamma &= +\dot{\theta}(\sin\beta\sin\gamma) - \dot{\beta}(\cos\gamma) \end{aligned}\right\} \qquad (2\text{-}25)$$

Adding the terms of Eq. (2-24) and (2-25) with $\overline{Z}$, Eq. (2-23) is rewritten as

$$\begin{aligned} \overline{\omega}_{V/F} = {} & [\dot{\gamma} - \dot{\phi}(\sin\beta\cos\theta) - \dot{\theta}(\cos\beta)]\hat{i}_\beta \\ & + [\dot{\phi}(\cos\beta\cos\gamma\cos\theta + \sin\gamma\sin\theta) - \dot{\theta}(\sin\beta\cos\gamma) - \dot{\beta}(\sin\gamma)]\hat{i}_V \\ & + [\dot{\phi}(-\cos\beta\sin\gamma\cos\theta + \cos\gamma\sin\theta) + \dot{\theta}(\sin\beta\sin\gamma) - \dot{\beta}(\cos\gamma)]\hat{i}_\gamma \end{aligned} \qquad (2\text{-}26)$$

---

* [M-2]and [M-4] denote the transformation matrices contained in Eq. (M-2) and (M-4).

By substituting the kinematic expressions for $\dot{\theta}$ and $\dot{\phi}$ which are given in Eq. (2-12) and (2-13) into Eq. (2-26), the $\dot{\theta}$ and $\dot{\phi}$ parameters are eliminated from Eq. (2-26) yielding:

$$\bar{\omega}_{V/F} = \left[\dot{\gamma} - \frac{V\cos\gamma}{R}\right]\hat{i}_\beta + \left[\frac{V}{R}(\sin\beta\sin\gamma\cos\gamma\tan\theta) - \dot{\beta}(\sin\gamma)\right]\hat{i}_V + \left[\frac{V}{R}(\sin\beta\cos^2\gamma\tan\theta - \dot{\beta}(\cos\gamma)\right]\hat{i}_\gamma \tag{2-27}$$

From the definition of the Relative Velocity Coordinate System,

$$\bar{V} = V\hat{i}_V \quad \text{and} \quad \left(\frac{d\bar{V}}{dt}\right)_{RVC} = \dot{V}\hat{i}_V \tag{2-28}$$

$\bar{\omega}_{V/F} \times \bar{V}$ can now be obtained

$$\bar{\omega}_{V/F} \times \bar{V} = \begin{vmatrix} \hat{i}_\beta & \hat{i}_V & \hat{i}_\gamma \\ \left[\dot{\gamma} - \frac{V\cos\gamma}{R}\right] & \left[\frac{V}{R}(\sin\beta\sin\gamma\cos\gamma\tan\theta) - \dot{\beta}(\sin\gamma)\right] & \left[\frac{V}{R}(\sin\beta\cos^2\gamma\tan\theta - \dot{\beta}(\cos\gamma)\right] \\ 0 & V & 0 \end{vmatrix}$$

By solving this determinant and adding the resulting expression $\dot{V}\hat{i}_V$, the expanded form of Eq. (2-20) is obtained; viz.,

$$\ddot{\bar{R}} = \left(\frac{d\bar{V}}{dt}\right)_{RVC} + \bar{\omega}_{V/F} \times \bar{V} = \left[\dot{\beta}V(\cos\gamma) - \frac{V^2}{R}(\sin\beta\cos^2\gamma\,\mathrm{Tan}\,\theta)\right]\hat{i}_\beta + [\dot{V}]\hat{i}_V + \left[\dot{\gamma}V - \frac{V^2}{R}(\cos\gamma)\right]\hat{i}_\gamma \tag{2-29}$$

The remaining terms of Eq. (2-21) which must be expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system each contain the vector $\bar{\omega}$. $\bar{\omega}$ is easily expressed in the Earth-Fixed Coordinate System as $\omega\hat{e}_3$. Use of the transformation matrix [M-4] yields $\bar{\omega}$ expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system.

$$\left.\begin{aligned} &\begin{pmatrix}\omega_\beta\\ \omega_V\\ \omega_\gamma\end{pmatrix} = [M\text{-}4]\begin{pmatrix}0\\0\\ \omega\end{pmatrix} \Longrightarrow \bar{\omega} = \omega_\beta\hat{i}_\beta + \omega_V\hat{i}_V + \omega_\gamma\hat{i}_\gamma \\ &\bar{\omega} = [-\omega(\sin\beta\cos\theta)]\hat{i}_\beta + [\omega(\cos\beta\cos\gamma\cos\theta + \sin\gamma\sin\theta)]\hat{i}_V + [\omega(-\cos\beta\sin\gamma\cos\theta + \cos\gamma\sin\theta)]\hat{i}_\gamma \end{aligned}\right\} \tag{2-30}$$

The term $\bar{\omega} \times \dot{\bar{R}}$ is the same as $\bar{\omega} \times \bar{V}$ by virtue of Eq. (2-2) and can readily be determined from:

$$\bar{\omega} \times \dot{\bar{R}} = \bar{\omega} \times \bar{V} = \begin{vmatrix} \hat{i}_\beta & \hat{i}_V & \hat{i}_\gamma \\ [-\omega(\sin\beta\cos\theta)] & [\omega(\cos\beta\cos\gamma\cos\theta + \sin\gamma\sin\theta)] & [\omega(-\cos\beta\sin\gamma\cos\theta + \cos\gamma\sin\theta)] \\ 0 & V & 0 \end{vmatrix}$$

Solving the determinant and simplifying yields:

$$\bar{\omega} \times \dot{\bar{R}} = [\omega V(\cos\beta\sin\gamma\cos\theta - \cos\gamma\sin\theta)]\hat{i}_\beta + [0]\hat{i}_V + [-\omega V(\sin\ \ \cos\theta)] \qquad (2\text{-}31)$$

The final term of Eq. (2-1) which must be expressed in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system is $\bar{\omega} \times (\bar{\omega} \times \bar{R})$. $\bar{R}$ is expressed in this system by application of the transformation matrix [M-2].

$$\begin{pmatrix} R_\beta \\ R_V \\ R_\gamma \end{pmatrix} = [M\text{-}2]\begin{pmatrix} 0 \\ 0 \\ R \end{pmatrix} \Longrightarrow \bar{R} = [0]\hat{i}_\beta + [R(\sin\gamma)]\hat{i}_V + [R(\cos\gamma)]\hat{i}_\gamma \qquad (2\text{-}32)$$

Then:

$$\bar{\omega} \times \bar{R} = \begin{vmatrix} \hat{i}_\beta & \hat{i}_V & \hat{i}_\gamma \\ [-\omega(\sin\beta\cos\theta)] & [\omega(\cos\beta\cos\gamma\cos\theta + \sin\gamma\sin\theta)] & [\omega(-\cos\beta\sin\gamma\cos\theta + \cos\gamma\sin\theta)] \\ 0 & [R(\sin\gamma)] & [R(\cos\gamma)] \end{vmatrix}$$

Solving the determinant yields:

$$\bar{\omega} \times \bar{R} = [\omega R(\cos\beta\cos\theta)]\hat{i}_\beta + [\omega R(\sin\beta\cos\gamma\cos\theta)]\hat{i}_V + [-\omega R(\sin\beta\sin\gamma\cos\theta)]\hat{i}_\gamma \qquad (2\text{-}33)$$

$$\bar{\omega} \times (\bar{\omega} \times \bar{R}) = \begin{vmatrix} \hat{i}_\beta & \hat{i}_V & \hat{i}_\gamma \\ [-\omega(\sin\beta\cos\theta)] & [\omega(\cos\beta\cos\gamma\cos\theta + \sin\gamma\sin\theta)] & [\omega(-\cos\beta\sin\gamma\cos\theta + \cos\gamma\sin\theta)] \\ [\omega R(\cos\beta\cos\theta)] & [\omega R(\sin\beta\cos\gamma\cos\theta)] & [-\omega R(\sin\beta\sin\gamma\cos\theta)] \end{vmatrix}$$

which when solved becomes:

$$\begin{aligned} \bar{\omega} \times (\bar{\omega} \times \bar{R}) = {} & [\omega^2 R(-\sin\beta\sin\theta\cos\theta)]\hat{i}_\beta \\ & + [\omega^2 R(-\sin\gamma\cos^2\theta + \cos\beta\cos\gamma\sin\theta\cos\theta)]\hat{i}_V \\ & + [\omega^2 R(-\cos\gamma\cos^2\theta - \cos\beta\sin\gamma\sin\theta\cos\theta)]\hat{i}_\gamma \end{aligned} \qquad (2\text{-}34)$$

The equations for the terms of Eq. (2-1) are Eq. (2-15), (2-17), (2-29), (2-31), and (2-34). These term equations express all terms of Eq. (2-1) in the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ coordinate system. Since Eq. (2-1) equates two vectors expressed in the same coordinate system, the vector components along like axes must be equal. Consequently, by equating the like components of the term equations remembering to include any required coefficients which were previously omitted (e.g., m, and 2), three scalar equations are obtained. By equating the $\hat{i}_\beta$ components and noting that $m = w/g_0$ (g_0 is the reference surface gravitational acceleration):

$$F_\beta - mg_\theta(\sin\beta) = \frac{W}{g_0}\left[\dot{\beta}V(\cos\gamma) - \frac{V^2}{R}(\sin\beta\cos^2\gamma\tan\theta) + 2\omega V(\cos\beta\sin\gamma\cos\theta - \cos\gamma\sin\theta) - \omega^2 R(\sin\beta\sin\theta\cos\theta)\right]$$

where F_β is defined in Eq. (2-15); rearranging yields:

$$\boxed{\dot{\beta} = \frac{1}{V(\cos\gamma)}\left[\frac{V^2}{R}(\sin\beta\cos^2\gamma\tan\theta) - g_\theta(\sin\beta) + \frac{F_\beta g_0}{W} + 2\omega V(\cos\gamma\sin\theta - \cos\beta\sin\gamma\cos\theta) + \omega^2 R(\sin\beta\sin\theta\cos\theta)\right]} \quad (2\text{-}35)$$

By equating the $\hat{i}_V$ components the second scalar equation is obtained.

$$F_V + \frac{W}{g_0}(g_\theta\cos\beta\cos\gamma - g_R\sin\gamma) = \frac{W}{g_0}\left[\dot{V} + \omega^2 R(-\sin\gamma\cos^2\theta + \cos\beta\cos\gamma\sin\theta\cos\theta)\right]$$

where F_V is defined by Eq. (2-15). By rearrangement

$$\boxed{\dot{V} = \left[g_\theta(\cos\beta\cos\gamma) - g_R(\sin\gamma) + \frac{F_V g_0}{W} + \omega^2 R(\cos\theta)(\sin\gamma\cos\theta - \cos\beta\cos\gamma\sin\theta)\right]} \quad (2\text{-}36)$$

The $\hat{i}_\gamma$ components yield the third scalar equation (in which F_γ is defined in Eq. (2-15))

$$F_\gamma + \frac{W}{g_0}(-g_\theta\cos\beta\sin\gamma - g_R\cos\gamma) = \frac{W}{g_0}\left[\dot{\gamma}V - \frac{V^2}{R}(\cos\gamma) - 2\omega V(\sin\beta\cos\theta) + \omega^2 R(-\cos\gamma\cos^2\theta - \cos\beta\sin\gamma\sin\theta\cos\theta)\right]$$

which when rearranged yields:

$$\boxed{\dot{\gamma} = \frac{V}{R}(\cos\gamma) + \left[\frac{1}{V}\right]\left[-g_\theta(\cos\beta\sin\gamma) - g_R(\cos\gamma) + \frac{F_\gamma g_0}{W} + 2\omega V(\sin\beta\cos\theta) + \omega^2 R(\cos\theta)(\cos\gamma\cos\theta + \cos\beta\sin\gamma\sin\theta)\right]} \quad (2\text{-}37)$$

Equations (2-35), (2-36), and (2-37) comprise the scalar kinetic equations of motion. These equations are the scalar representation of Eq. (2-1) which describes the motion of an aerospace vehicle in a gravitational field with harmonic terms under the influence of applied forces, and are expressed in terms of local position and relative velocity parameters. The simultaneous solution of these kinetic equations together with the kinematic equations (Eq. (2-11), (2-12), and (2-13)) completely define the trajectory since this solution yields the time history of the state variables R, V, γ, β, θ, and ϕ. The other trajectory parameters can be readily obtained as functions of these six state variables.

2.2.2.4 <u>Supplementary Trajectory Parameter Differential Equations.</u> In addition to the six trajectory parameters which are obtained by solution of the kinematic and kinetic differential equations of motion, there are other trajectory-related parameters of interest which are also obtained by solution of their differential equations. These parameters include the current weight, ideal propulsive velocity, gravity relative velocity loss, aerodynamic relative velocity loss, thrust misalignment relative velocity loss, and a heating parameter.

<u>Weight.</u> The vehicle weight (W) is obtained by integrating the time rate of change of weight which is defined by various models. The total time rate of change of weight ($\dot{W}$) is composed of three terms; the $\dot{W}$ term which is derived as part of the SIMPO propulsion options ($\dot{W}_{SIMPO}$), the $\dot{W}$ term which is derived from the Propulsion Table options ($\dot{W}_{TABLE}$), and the $\dot{W}$ term which accounts for any GIMAC* control flow rates ($\dot{W}_{GIMAC}$). These three $\dot{W}$ terms are independent of each other.

*Gas injection maneuvering and control

Both the $\dot{W}_{SIMPO}$ and the $\dot{W}_{TABLE}$ terms can be directly defined by the appropriate input parameters. This is true even if there is no corresponding thrust. These parameters can also be derived from the currently defined reference thrust (T_{SIMPO} or T_{TABLE}) and reference specific impulse (I_{SIMPO} or I_{TABLE}) values of their particular option by use of the relationship between weight flow rate, thrust, and specific impulse:

$$\dot{W}_{SIMPO} = \frac{T_{SIMPO}}{I_{SIMPO}}$$

$$\dot{W}_{TABLE} = \frac{T_{TABLE}}{I_{TABLE}} \tag{2-38}$$

If control by gas ejection (GIMAC control) is to be provided, it is necessary to account for the associated weight expenditure rate. Assuming that the vehicle is flying at or near the trim angle of attack, this GIMAC weight flow rate is primarily dependent on the aerodynamic normal force coefficient (C_ζ), the dynamic pressure (q), the Mach number (M), the relative velocity (V), and four GIMAC coefficients (k_1, k_2, k_3, and k_4) and is expressed as:

$$\dot{W}_{GIMAC} = 2C_\zeta q\,(k_1M^3 + k_2M^2 + k_3M + k_4)/V \tag{2-39}$$

Each weight flow rate term is assumed to be positive if it, taken by itself, would cause the weight of the vehicle to decrease with increasing time. None, any, or all three $\dot{W}$ parameters can be used to define the total time rate of change in vehicle weight. For example, if all three parameters are used:

$$\dot{W} = \dot{W}_{SIMPO} + \dot{W}_{TABLE} + \dot{W}_{GIMAC} \tag{2-40}$$

Ideal Propulsive Velocity and Associated Velocity Losses. Four velocity parameters are useful for trajectory analyses; these are: 1) the propulsive ideal velocity, 2) the relative velocity loss due to gravity, 3) the relative velocity loss due to aerodynamic forces, and 4) the relative velocity loss due to thrust misalignment.

The propulsive ideal velocity is that velocity which would be achieved by the application of only the thrust forces from the initiation of the simulated trajectory to the current time. In calculating this ideal velocity, it is assumed that there are no gravity or aerodynamic forces, and that at any time the total effective thrust vector is aligned with the relative velocity vector. The effects of the atmospheric pressure on the total thrust are included in the effective thrust. Consequently the ideal velocity defined here differs from the ideal velocity given by the logarithmic rocket equation since this equation assumes a constant specific impulse.

The total propulsive ideal velocity (ΔV_{ID}) at time t is given by:

$$\Delta V_{ID} = \int_{t_0}^{t} \left(\frac{T g_0}{W}\right) dt \tag{2-41}$$

where: g_0 is the reference surface gravitational acceleration.

t is the current time.

T is the total current effective thrust acting on the vehicle in line with the vehicle roll axis $\hat{\xi}$.

t_0 is the time at the initiation of the trajectory simulation.

W is the total current weight of the vehicle.

The relative velocity loss due to gravity (ΔV_g) is the velocity difference from the ideal velocity which results from the presence of a gravitational field. Contributions to the gravity losses occur whenever the gravitational acceleration vector ($\bar{g}$) has a net component which is parallel to the relative velocity vector. The net gravitational acceleration component in the $-\bar{V}$* direction is obtained by transforming the $\bar{g}$ vector from the Local Coordinate System to the Relative Velocity Coordinate System using Eq. (M-2). This component is the negative of the coefficient of the $\hat{i}_V$ axis. Noting the result which is given in Eq. (2-17) the gravity loss is expressed by:

$$\Delta V_g = \int_{t_0}^{t} (g_R \sin \gamma - g_\theta \cos \mathcal{L} \cos \gamma)\, dt \tag{2-42}$$

*Taken as negative by convention.

The velocity difference from the ideal velocity which is due to the aerodynamic forces is called the aerodynamic relative velocity loss (ΔV_A). This loss occurs whenever there are aerodynamic forces acting on the vehicle which have net components which are parallel to the relative velocity vector. The net aerodynamic force in the $\bar{V}$ direction is obtained by transforming the total aerodynamic force vector ($\bar{F}_{AERO}$) from the Vehicle Coordinate System to the Relative Velocity Coordinate System by means of Eq. (M-1). This component is the coefficient of the $\hat{i}_V$ axis. By noting the results of Eq. (2-15), where F_ξ which contains propulsive forces is replaced by the axial aerodynamic force (F_A) which does not contain propulsive forces, the aerodynamic loss is given by:

$$\Delta V_A = \int_{t_0}^{t} \left\{ \frac{\left[-(\cos\lambda \cos\alpha)\, F_A + (\sin\lambda \cos\alpha)\, F_\eta - (\sin\alpha)\, F_\zeta \right] g_0}{W} \right\} dt \qquad (2\text{-}43)$$

The relative velocity loss due to thrust misalignment (ΔV_M) is the velocity decrement from the ideal velocity which results when the total effective thrust vector has net components which are not parallel to the relative velocity vector. This thrust "loss" along the $\hat{i}_V$ axis is the difference between the total effective thrust magnitude and its component along the $\hat{i}_V$ axis. This component is readily obtained by application of Eq. (M-1) to the thrust magnitude (T) to transform it from the Vehicle Axes to the Relative Velocity Axes and is equivalent to the resulting component along the $\hat{i}_V$ axis. Applying the results of Eq. (2-15), where F_ξ which includes aerodynamic forces is replaced by T which does not include aerodynamic forces, the thrust misalignment velocity loss is expressed as:

$$\Delta V_M = \int_{t_0}^{t} \left(\frac{T\, g_0\, (1 - \cos\lambda \cos\alpha)}{W} \right) dt \qquad (2\text{-}44)$$

After initial consideration, one might assume that:

$$V = \Delta V_{ID} - \Delta V_g - \Delta V_A - \Delta V_M \qquad \text{(NOT TRUE)}$$

This is not true although in some cases it is nearly true. Examination of Eq. (2-36), which defines, the time derivative of the relative velocity, indicates that the gravity loss is included in terms of its time derivative as the first two right-hand terms of the equation. By expanding F_V of the third right-hand term of Eq. (2-36) by means of Eq. (2-15) and noting that $F_\xi = F_A + T$, it can be seen that the total ideal velocity minus the thrust misalignment plus the aerodynamic velocity losses are represented as their time derivatives in this third right-hand term. This leaves the fourth right-hand term of Eq. (2-36) unaccounted for in terms of velocity losses. This term is actually the Coriolis acceleration component (it has a corresponding velocity loss (ΔV_{COR})) in the direction of the instantaneous relative velocity vector. For most trajectories encountered in which the velocity losses are useful, this term is small* when compared with the total ideal velocity. The complete relative velocity — ideal velocity relationship becomes:

$$V = \Delta V_{ID} - \Delta V_g - \Delta V_A - \Delta V_M - \Delta V_{COR} \qquad (2\text{-}45)$$

when the algebraic sign is in accordance with Eq. (2-15), (2-36), (2-41), (2-42), (2-43), and (2-44).

Aerodynamic Heating Parameter. The heat flux (H_F) is defined to be the product of the dynamic pressure (q) and the relative velocity (V). The heating parameter (H_p) at time (t) is the time integral of the heat flux. These two relationships are:

$$H_F = qV$$

$$H_p = \int_{t_0}^{t} H_F \, dt = \int_{t_0}^{t} q V \, dt \qquad (2\text{-}46)$$

where t_0 is the time at the initiation of the trajectory simulation.

---

* For example, for a typical ascent to orbit trajectory with a $\Delta V_{ID} \approx 30{,}000$ ft/sec, 20 ft/sec $< \Delta V_{COR} <$ 200 ft/sec.

2.2.2.5 Trajectory Solution Singularities. There are some trajectory situations which result in an inability to define some of the kinematic and/or kir tic state variables; these require special logic to insure the correct solution, or to yield desirable flight control histories.

Undefined Cases. Examination of the state kinematic and kinetic equations of motion (Eq. (2-11), (2-12), (2-13), (2-35), (2-36), and (2-37)) indicate that the solvability of all except Eq. (2-11) which defines $\dot{R}$ depends on the values of some of the parameters which appear in the terms. The parameters whose values can present difficulties are the geocentric radius magnitude (R), the relative velocity magnitude (V), the vehicle weight (W), the relative flight path angle (γ), and the geocentric latitude (θ). Since R is never zero no special provisions for R are required.

Normally the relative velocity magnitude is greater than zero, however if the vehicle has an inertial velocity with a due east inertial azimuth, a zero inertial flight path angle, and a magnitude such that the vehicle has no motion with respect to the local meridian the relative velocity is zero. This condition most often occurs either at liftoff with a zero initial velocity or in synchronous equatorial orbit. The equations which become critical as V approaches zero are those which define the time derivatives of the relative azimuth (Eq. (2-35)) and the time derivative of the relative flight path angle (Eq. (2-37)). The lift-off case, which occurs only at the initiation of a trajectory simulation, is easily handled by assuming initial β and γ values and assuming that the β time derivative ($\dot{\beta}$) is zero until after a specified time (provided by input parameter "ATQ"). The synchronous equatorial case and all other cases which might arise are handled by assuming that the time derivatives $\dot{\gamma}$ and $\dot{\beta}$ (Eq. (2-37) and (2-35) respectively) are computed by substituting a specified (provided by input parameter "VELQ") minimum relative velocity magnitude in the denominator of these two equations whenever the specified value of V is less than this specified minimum. This has the effect of smoothing the values of these functions by not permitting their values to become too large in magnitude, while providing a reasonable starting value for these derivatives

2 A

if the zero V condition is later passed. The physical significance of these parameters on the resulting trajectory is minor since when the velocity is near zero, its direction and magnitude do not appreciably affect position (Eq. (2-11), (2-12), and (2-13)). When operating near these conditions, care must be exercised to insure that the integration tolerances are sufficiently small to prevent V from becoming negative. From the definition of V, this condition should theoretically never occur, however the integration process uses polynomials to predict the succeeding values of the integrated variables, consequently the tolerances must be such to prevent the polynomial determined value of V from becoming negative.

The vehicle weight (W) should never be zero or negative; however this can occur when trying to achieve a simulation section termination condition which is unreasonable for the modeled vehicle and the trajectory it is flying. This problem can be avoided by use of the back-up time termination procedure (see Section 4.2.3.5, Program Control) which provides a secondary simulation section termination capability. In this case, if the normal simulation section termination conditions are not met by the time which is defined by the back-up time, the simulation section is terminated. The back-up time, then, should be determined on the basis of allowing sufficient time for the required propellant expenditure plus whatever margin of time is necessary to prevent interference with the normal simulation section termination procedure and the determination of any required iteration end conditions.

The kinetic equation for $\dot{\beta}$ (Eq. (2-35)) becomes critical for γ values near either +90 or -90 degrees. Normally this only occurs at lift-off and during the subsequent vertical rise for ascent trajectories. Physically, of course, the azimuth β has no meaning under these circumstances. For the initial lift-off case, an initial launch azimuth value is assumed and the time rate of change of β ($\dot{\beta}$) given by Eq. (2-35) is assumed to be zero for $|\gamma|$ values above a specified value provided by input parameter "AGQ". Additionally, the time delay parameter ("ATQ") which is used to delay computation of $\dot{\beta}$ for low initial relative velocities is also applicable in this situation.

Should this condition arise at any other time in the trajectory, the $\dot{\beta}$ equation is automatically set to zero whenever either $|\gamma|$ is greater than "AGQ" or $|\gamma|$ is within 0.00005 degree of 90 degrees. It is noted that under those conditions, $\dot{\theta}$ and $\dot{\phi}$ are identically zero (Eq. (2-12) and (2-13)).

At either central body rotational axis pole, both the relative azimuth (β) and the longitude ϕ (Eq. (2-35) and (2-13), respectively) are undefined. From a physical point of view, these parameters have no definition under these conditions; however certain assumptions are made which result in realistic definitions for β and ϕ. If the position results in a geocentric latitude (θ) such that $|\theta|$ is within 0.0005 degree of 90 degrees. these equations which define $\dot{\beta}$ and $\dot{\phi}$ are set at zero. This means that the previously determined values of β and ϕ (either from the previous integration step or from the initial values if the trajectory is initiated at one of the poles) are assumed at the pole in question. After a pole is crossed, the integration procedure would be expected to predict about the same values (as before the crossing) for β and ϕ (in the absence of any drastic turning maneuver) and a value of θ the magnitude of which is greater than 90 degrees, however when this condition arises, these parameters are reset to correspond to the physical situation by the following scheme:

θ: $\theta = 180° - \theta_0$ if at the north pole (θ_0 near +90°)

$\theta = -180° - \theta_0$ if at the south pole (θ_0 near -90°)

β: $\beta_0 + 180°$ *

ϕ: $\phi_0 + 180°$ *

where the 0 subscript here denotes the polynomial-predicted value after the pole was crossed and before the logic just described has been applied.

Rotational Rate of the Central Body. In the derivation of the vector kinetic equation (Eq. (2-1)), it was assumed that the inertial frame had its origin at the center of the central body and that the orientation was fixed with respect to the stars in our galaxy.

* β and ϕ can be adjusted such that both β and ϕ are between 0 and 360 degrees if required (See Section 4.2.3.5 Program Control for the "PRIF(K)" option).

The rotational rate of the Earth-Fixed Coordinate System about the Inertial System is then the rotational rate of the central body (ω) and should be defined by using the sidereal day (approximately 86164.091* mean solar seconds for the earth) rather than the mean solar day (exactly 86400* mean solar seconds for the earth).** If another ω were used, then the relative parameters would be defined with respect to a coordinate system which rotates about the I_3 axis at a rate other than that of the central body. This characteristic has specific applications to some analyses where the results are assumed to be independent of the trajectory direction, e.g., a non-rotating earth solution. In this case, the rate ω is assumed to be zero. Another application requiring a special rotating system is rendezvous analyses where it is necessary to determine the spacecraft's trajectory with respect to a satellite and the satellite orbit. In simulating this case the $e_1 - e_2$ plane of the "Earth-Fixed" Coordinate System is coplanar with the satellite orbital plane and the origin of this system remains at the center of the attracting body. Consequently the I_3 axis (and the e_3 axis which is colinear with I_3) becomes the axis of the satellite orbital motion. The "Earth-Fixed" Coordinate System is given a rotational rate so that it rotates with the angular motion of the satellite.

2.2.2.6 Specified Time Rate of Change of Relative Flight Path Angle. For some types of flight it is necessary to be able to control the altitude, the radius magnitude, or the relative flight path angle time history. Typical examples are atmospheric entry and subsequent aerodynamic flight of a maneuverable lifting vehicle, or the relatively low-altitude flight of an air-launched missile which is subjected to both aerodynamic and propulsive forces. In an actual flight, this control is accomplished by by modulation of the vehicle attitude to obtain the necessary propulsive and/or aerodynamic force components along the geocentric radius vector. For this reason, and noting that the kinematic equations define only the geometry of motion rather than the motion resulting from external forces, it is necessary to use the kinetic equations of motion which contain the external forces to control the resulting trajectory. Inspection of the kinetic equations of motion (Eq. (2-35), (2-36), and (2-37)) indicates that

---

* Reference 5

** Use of this ω (0.0041780746*deg/sec for the earth) results in the relative parameters being defined with respect to the surface of the central body.

the only suitable equation for this purpose is Eq. (2-37), which defines $\dot{\gamma}$.* Theoretically there are three modes which can be used to control the F_γ term of this equation. These are: bank angle modulation, pitch angle of attack modulation, and yaw angle of attack modulation. Of course any two or all of these may be used simultaneously for control; however, in most circumstances either of the first two modes used independently provides excellent control of the relative flight path angle. Consequently only the bank angle and pitch angle of attack modulation modes are provided in the program; and only one of these two modes may be used at a time, that is only one may be used during any simulation section.

The technique which is used to effect control of the altitude, radius magnitude, or relative flight path angle time history, is to assume a constant time rate of change of relative flight path angle and then solve for the required bank angle (σ) or pitch angle of attack (α). Equation (2-37) can be rearranged as:

$$F_\gamma = \left(\frac{W}{g_0}\right)\Bigg[g_\theta(\sin\gamma\cos\beta) + g_R(\cos\gamma) - 2\omega_e V(\sin\beta\cos\theta) - \omega_e^2 R(\cos\theta)(\cos\gamma\cos\theta + \sin\gamma\cos\beta\sin\theta) + V\left\{\frac{\dot{\gamma}_s}{RAD} - \frac{V}{R}(\cos\gamma)\right\}\Bigg] \tag{2-47}$$

where:

$\dot{\gamma}_s$ is the specified time rate of change of the relative flight path angle, in degrees/second.

RAD is the radian to degree conversion factor; 57.2957795 degrees/radian.

The F_γ term of Eq. (2-15) is defined in terms of the three $\hat{\xi}-\hat{\eta}-\hat{\zeta}$ components of the applied force (F_ξ, F_η, F_ζ) and the roll angle (σ), the pitch angle of attack (α), and the yaw angle of attack (λ). The aerodynamic force components of F_ξ and F_ζ (F_A and F_ζ) are dependent on Mach number and α while the aerodynamic force component F_η is dependent on Mach number and λ.

* It is noted that control of γ is not really equivalent to controlling the altitude because of the oblate surface geometry.

If turning is required during any phase of flight in which a specified time rate of change of relative flight path angle is specified, then it is necessary to use bank angle modulation. The bank angle (σ_s) which is required to maintain a constant $\dot{\gamma}_s$ can be readily obtained by assuming that α is independent of σ, that $\lambda = 0$, and that $F_\eta = 0$. These assumptions are nearly the case for most applications. Since none of F_ξ, F_η, or F_ζ are dependent on σ, then the F_γ term of Eq. (2-15) reduces to:

$$F_\gamma = (\sin\alpha \cos\sigma) F_\xi + (-\cos\alpha \cos\sigma) F_\zeta \tag{2-48}$$

from which the required roll angle can be obtained as:

$$\sigma_s = \text{ARCCOSINE} \left[\frac{F_\gamma}{\left| (\sin\alpha) F_\xi - (\cos\alpha) F_\zeta \right|} \right] \tag{2-49}$$

Eq. (2-49) is symmetric about $\sigma = 0$, that is either a right or a left bank angle (σ_s) can be obtained from this equation; consequently the algebraic sign* of the bank angle must be specified in order to completely define σ_s. In addition, if the required F_γ defined by Eq. (2-47) is greater than what can be obtained from acceptable σ values (α is assumed to be independently specified) then there is no acceptable solution to Eq. (2-49). Indeed, the magnitude of the denominator of Eq. (2-49) can never be less than $|F_\gamma|$. Should this denominator equal F_γ, then σ_s is identically zero. The adopted procedure is to specify a minimum (maximum)† σ_{LIM}; if the computed σ_s value is below (above)† σ_{LIM}, then σ_{LIM} is assumed instead of σ_s. Upon reaching σ_{LIM}, the specified $\dot{\gamma}_s$ can no longer be met by bank angle modulation for the current trajectory conditions (α and Mach number); it is however possible to achieve $\dot{\gamma}_s$ later on if the required trajectory conditions are met. Finally, σ_{LIM} must have the same sign as the initial σ since generally the σ_s function is monotonic while for constant trajectory parameters (α and Mach number) the F_γ given by Eq. (2-48) passes through a maximum at $\sigma = 0$.

---

* Positive for right bank angles, negative for left.

† Algebraic sense; the condition specified in parentheses is applicable to left bank angles.

Pitch angle of attack modulation is used to maintain a constant $\dot{\gamma}_s$ when an independently specified bank angle (σ) and/or yaw angle (λ) is required. Unfortunately there is no simplified solution such as Eq. (2-49) for the pitch angle of attack (α_s) required to maintain a constant $\dot{\gamma}_s$. In general F_ζ and F_ζ are dependent on α and in some cases of α modulation, non-zero values of both σ and λ might be required. To solve for α_s, a Newton-Raphson iteration is performed on the α in the equation which defines F_γ in terms of σ, α, λ, F_ξ, F_η, and F_ζ — Eq. (2-15). At each iteration step, the F_ξ and F_ζ terms are evaluated for the current α value. This solution, thus, requires only that σ and λ be defined independently of α and that σ, λ, F_ξ, F_η, and F_ζ have reasonable values. Pitch angle of attack modulation has bounds which are usually determined from the vehicle characteristics. Since the expression for F_γ (Eq. (2-15)) is non-symmetric about $\alpha = 0$, no special consideration other than strict adherence to algebraic sign convention is required for negative α values. α is bounded from above and below and these α limits (α_{MIN} and α_{MAX}) must be specified. If the iteration calls for an α outside the bounds, the value of the nearest α limit is assumed and the trajectory simulation is continued. This use of an α limit means that α modulation for the trajectory conditions being considered cannot provide the required F_γ (from Eq. (2-47)) to maintain the constant $\dot{\gamma}_s$. α modulation will be resumed automatically as soon as the trajectory conditions permit it.

2.2.2.7 Solution by Numerical Integration. It is necessary to solve the six simultaneous differential equations of motion (Eq. (2-11), (2-12), (2-13), (2-35), (2-36), and (2-37)) in order to define the vehicle trajectory. It is generally not possible to solve these equations and the six supplementary trajectory parameter differential equations (Eq. (2-40), (2-41), (2-42), (2-43), (2-44), and (2-46) in closed form, consequently, it is necessary to employ a numerical procedure to integrate these twelve equations.

An integration scheme which is simple and precise, yet economical, for this problem is that suggested by R. H. Merson (Reference 6) in 1957. This process, known as the Kutta-Merson Integration Technique, is a fourth-order Runge-Kutta type with provision for an automatic integration step interval adjustment. This technique is simple to initiate, simple to terminate, and does not require storage of parameters computed during previous steps. A summary description of this technique follows.

Let each of the N differential equations be represented:

$$\frac{dy^{(k)}}{dx} = f_k \left(x,\ y^{(1)},\ y^{(2)},\ y^{(3)},\ \ldots\ldots\ y^{(N)} \right) \tag{2-50}$$

where

x is the independent variable

$y^{(k)}$ is the k-th simultaneously dependent variable

f_k is the k-th functional relationship which defines the derivative of $y^{(k)}$ with respect to x.

It is emphasized that f_k need not be analytically expressed; it can include terms defined by tables, and/or specific input, and/or analytic function (e.g., F_β, F_V, and F_γ of Eq. (2-35), (2-36), and (2-37), respectively). For this application, whenever f_k is evaluated, it is necessary to evaluate the vehicle and environmental characteristics for each value of x which is considered.

Let h be the integration interval step and let the subscript of $y_j^{(k)}$ denote the j-th predicted value of $y^{(k)}$. The zero subscript of x_0 and $y_0^{(k)}$ denotes the values of x and $y^{(k)}$ at the initiation of the integration. The Kutta-Merson process (Reference 7) for each dependent variable is shown on the following page.

$$\left.\begin{aligned}
y_1^{(k)} &= y_0^{(k)} + \frac{1}{3} h f_k \left(x_0, y_0^{(1)}, y_0^{(2)}, \ldots . y_0^{(N)}\right) \\
y_2^{(k)} &= y_0^{(k)} + \frac{1}{6} h f_k \left(x_0, y_0^{(1)}, y_0^{(2)}, \ldots . y_0^{(N)}\right) + \frac{1}{6} h f_k \left(\left[x_0 + \frac{1}{3} h\right], y_1^{(1)}, y_1^{(2)}, \ldots . y_1^{(N)}\right) \\
y_3^{(k)} &= y_0^{(k)} + \frac{1}{8} h f_k \left(x_0, y_0^{(1)}, y_0^{(2)}, \ldots . y_0^{(N)}\right) + \frac{3}{8} h f_k \left(\left[x_0 + \frac{1}{3} h\right], y_2^{(1)}, y_2^{(2)}, \ldots . y_2^{(N)}\right) \\
y_4^{(k)} &= y_0^{(k)} + \frac{1}{2} h f_k \left(x_0, y_0^{(1)}, y_0^{(2)}, \ldots . y_0^{(N)}\right) - \frac{3}{2} h f_k \left(\left[x_0 + \frac{1}{3} h\right], y_2^{(1)}, y_2^{(2)}, \ldots . y_2^{(N)}\right) \\
&\quad + 2 h f_k \left(\left[x_0 + \frac{1}{2} h\right], y_3^{(1)}, y_3^{(2)}, \ldots . y_3^{(N)}\right) \\
y_5^{(k)} &= y_0^{(k)} + \frac{1}{6} h f_k \left(x_0, y_0^{(1)}, y_0^{(2)}, \ldots . y_0^{(N)}\right) + \frac{2}{3} h f_k \left(\left[x_0 + \frac{1}{2} h\right], y_3^{(1)}, y_3^{(2)}, \ldots . y_3^{(N)}\right) \\
&\quad + \frac{1}{6} h f_k \left(\left[x_0 + h\right], y_4^{(1)}, y_4^{(2)}, \ldots . y_4^{(N)}\right)
\end{aligned}\right\} \quad (2\text{-}51)$$

where $y_5^{(k)}$ is the accepted value of $y^{(k)}$ at the end of the integration step $(x + h)$. Each group of $y_j^{(k)}$ is evaluated simultaneously. It is noted that f_k is only evaluated five times during each integration step regardless of the number of dependent variables. If the interval h is such that $f_k\ (x, y^{(1)}, y^{(2)}, \ldots y^{(N)})$ can be represented by the linear approximation

$$f_k\left(x, y^{(1)}, y^{(2)}, \ldots . y^{(N)}\right) = Ax + B^{(1)} y^{(1)} + B^{(2)} y^{(2)} + \ldots . B^{(N)} y^{(N)} + C$$

for $x_0 \leq x \leq x_0 + h$ and where A, $B^{(k)}$, and C are constants, then it can be shown (Reference 10) that the error of $y_4^{(k)}$ and $y_5^{(k)}$ is

$$\left.\begin{aligned}
\text{ERROR IN } y_4^{(k)} &= -\frac{1}{120} h^5 \left(\frac{d^5 y^{(k)}}{dx^5}\right) \\
\text{ERROR IN } y_5^{(k)} &= -\frac{1}{720} h^5 \left(\frac{d^5 y^{(k)}}{dx^5}\right)
\end{aligned}\right\} \quad (2\text{-}52)$$

From Eq. (2-52) it can be seen that the error in $y_5^{(k)}$ is 1/6 the error in $y_4^{(k)}$, consequently, a good estimate (Reference 7) for the error (ϵ (k)) in the accepted value of $y^{(k)}$ is:

$$\epsilon (k) = \frac{1}{5} \left| y_4^{(k)} - y_5^{(k)} \right| \tag{2-53}$$

The error estimate given by Eq. (2-53) for each dependent variable can be used to either expand or contract the integration step size. It is noted that the accuracy of the predicted $y_5^{(k)}$, is better for smaller values of h, consequently, it is reasonable to decrease the step size in the event that ϵ (k) is too large for any dependent variable(s) ($y^{(k)}$). If ϵ(k) is sufficiently small for all dependent variables ($y^{(k)}$) then the step size can be increased. This is exactly what is done. Dual sets of integration tolerances are specified for each dependent variable ($y^{(k)}$). The first set consists of upper tolerances (TOL_1 (k)) each one of which is approximately equal to the maximum acceptable error in the k-th dependent variable over an integration step. If ϵ (k) is larger than TOL_1(k) for any $y^{(k)}$, then the integration step size is decreased by a contraction coefficient (G_1) and the integration is restarted from the previous successfully integrated values. The second set of tolerances consist of the lower tolerances (TOL_2(k)). These lower tolerances are selected for each $y^{(k)}$ such that if each TOL_2 (k) is satisfied, expansion of the integration step size by a specified expansion coefficient G_2 will result in meeting the upper tolerances during the subsequent step. Table 2-1 shows these data in summary form.

Contraction and expansion coefficient values of 0.5 and 2.0 for G_1 and G_2, respectively; TOL_1(k) values selected as the total acceptable error at the end of the simulated trajectory divided by the total expected number of integration steps; and TOL_2 (k) values selected as 1/10 the corresponding TOL_1 (k) value have been used successfully for a wide range of types of trajectory simulations which have been determined with GTSM. Successfully used values of these patameters are internally compiled in GTSM; however, all (with the exception of G_1) can be input with a different value in any or all simulation sections.

Table 2-1. Summary of Error Conditions

Error Condition	Number of $y^{(k)}$ for Which Condition Exists	Effect on h and Integration Procedure
$\epsilon(k) < TOL_2(k)$ (lower tolerance)	All	The current integration step is successful and h is expanded to G_2h for starting the next integration step
$\epsilon(k) > TOL_1(k)$ (upper tolerance)	One or More	h is contracted to G_1h and the current integration step is reattempted
$\epsilon(k) < TOL_1(k)$ $\epsilon(k) > TOL_2(k)$	All One or More	The current integration step is successful and h is unchanged for starting the next integration step

To initiate integration it is necessary to know the values of x_0, $y_0^{(k)}$, h_0, G_1, G_2, $TOL_1(k)$, $TOL_2(k)$, and the minimum acceptable integration step size (h_{MIN}). The values of x_0 and $y_0^{(k)}$ are usually defined either from the initial conditions input for the first simulation section or from the values determined at the end of the previous simulation section. A simulation section is defined as any segment of the simulated trajectory in which the integration process is continuous; a section begins with the initiation of an integration process and ends with termination of that particular integration process. Each time that f_k is evaluated, any parameters* which are used to define terms in f_k must, of course, also be defined.

Termination of integration in a simulation section occurs either by direction or by error. A simulation section termination parameter is defined by input together with its termination value, direction, and tolerance. The termination direction indicates whether the value of the termination parameter should be increasing or decreasing at termination. When the integrated value of the termination parameter passes through its specified section termination value in the correct direction during an integration step, the value of the integration step size is iterated on by using the Newton-Raphson iteration technique until the integrated value of the termination parameter

* These parameters are known as the necessary trajectory parameters.

is within the termination tolerance of the termination value. The integration step is reattempted for each Newton-Raphson prediction. An auxiliary integration termination procedure is also used. This procedure, which is called back-up time termination, occurs automatically whenever the current time exceeds the specified (by program input) back-up time and the normal integration termination iteration procedure has not already been initiated. In the event that an error occurs when evaluating any term of f_k which requires a modeling table, the integration process is terminated with an appropriate diagnostic. If the integration step size contraction procedure results in requiring an integration step size which is less than the input minimum acceptable integration step size, a diagnostic is printed. Under these circumstances, integration termination is optional. If continuance of the integration is selected, the most recent reduced integration step size which is not less than this minimum step size is assumed. In this case, each integration step which fails to meet the integration tolerance is accompanied with an error magnitude for each of the twelve state variables listed in Table 2-2.

2.2.2.8 Trajectory Parameters. A trajectory parameter as used herein is a parameter which describes the vehicle, its motion, or its environment and is computed at least once during each integration step. There are four principal categories of trajectory parameters which will be subsequently elaborated upon; these are: 1) the state variables, 2) the necessary parameters, 3) the after-the-fact parameters, and 4) the optional after-the-fact parameters.

State Variables. The state variables as defined in Table 2-2 are the twelve parameters which are determined by numerical integration. These parameters have been defined previously but are listed in Table 2-2 for convenience.

Necessary Parameters. The necessary trajectory parameters, are those parameters which must be defined in order to determine the coefficients and terms of the twelve differential equations which are integrated. These necessary parameters must be evaluated at least five* times per integration step and, consequently, in order

* If the step size is contracted, these parameters must be evaluated five times per contraction. See Section 2.2.2.7 (Solution by Numerical Integration).

Table 2-2. State Variables

Symbol	Parameter	Program Internal Parameter Designation
R	Geocentric radius	Z(1)
V	Earth Relative Velocity	Z(2)
γ	Earth Relative Flight Path Angle	Z(3)
β	Earth Relative Azimuth Measured Clockwise from North	Z(4)
θ	Geocentric Latitude, Positive in the Northern Hemisphere	Z(5)
ϕ	Geocentric Longitude, Positive to the East of the Greenwich Meridian	Z(6)
W	Current weight of the Vehicle	Z(7)
ΔV_{ID}	Cumulative Propulsive Ideal Velocity	Z(17)
ΔV_g	Cumulative Gravity Relative Velocity Loss	Z(18)
ΔV_A	Cumulative Aerodynamic Relative Velocity Loss	Z(19)
ΔV_M	Cumulative Thrust Misalignment Relative Velocity Loss	Z(20)
H_p	Cumulative Heating Parameter	Z(39)

to eliminate redundancy, they are computed separately from the so-called after-the-fact trajectory parameters which are not required in defining the terms of the twelve differential equations. These necessary parameters are listed below in the order that they appear in Section 5.1.2.1 (Regular Output Parameter) together with their definitions. The corresponding program internal designation appears in parentheses to the right of each parameter symbol and the units appear in parentheses at the end of each definition.

<u>Time</u>

t (T) The absolute time which is measured from a reference time. The reference time is defined by specifying the time (t_0) at the initiation of the trajectory simulation (start of simulation section 1). If t_I is the current time from the initiation of simulation section 1, then

$$t = t_0 + t_I \qquad \text{(sec)} \qquad (2\text{-}54)$$

t_R (TR) The section relative time which is the current time from the initiation of the current simulation section. (sec)

Position

h (Z(8)) The local altitude above the surface of the oblate central body.

$$h = R - R_s \qquad \text{(ft)} \qquad (2\text{-}55)$$

Acceleration

F_η (Z(24)) The aerodynamic yaw force which is directed along the standard pitch axis ($\hat{\eta}$ axis). The algebraic sign is in accordance with the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ coordinate system which is defined in Section 2.2.2.1 (Positive to the Right).

$$F_\eta = C_\eta \, q \, A_\eta \qquad \text{(lb)} \qquad (2\text{-}56)$$

F_A (Z(25)) The aerodynamic axial force which is directed along the standard roll axis ($\hat{\xi}$ axis). The algebraic sign is in accordance with the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ coordinate system which is defined in Section 2.2.2.1 (Positive Forward).

$$F_A = C_A \, q \, A_A \qquad \text{(lb)} \qquad (2\text{-}57)$$

F_ξ (Z(28)) The total applied force which is directed along the standard roll axis ($\hat{\xi}$ axis). The algebraic sign is in accordance with the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ coordinate system which is defined in Section 2.2.2.1 (Positive Forward).

$$F_\xi = T + F_A \qquad \text{(lb)} \qquad (2\text{-}58)$$

F_ζ (Z(26)) The aerodynamic normal force which is directed along the standard yaw axis ($\hat{\zeta}$ axis). The algebraic sign is in accordance with the $\hat{\xi} - \hat{\eta} - \hat{\zeta}$ coordinate system which is defined in Section 2.2.2.1 (Positive downward).

$$F_\zeta = C_\zeta \, q \, A_\zeta \qquad (2\text{-}59)$$

ℓ_η (Z(29)) The aerodynamic yaw load factor which is directed along the η axis.

$$\ell_\eta = F_\eta / W \qquad (g) \qquad (2\text{-}60)$$

ℓ_A (Z(30)) The aerodynamic axial load factor which is directed along the ξ axis.

$$\ell_A = F_A / W \qquad (g) \qquad (2\text{-}61)$$

ℓ_ζ (Z(31)) The aerodynamic normal load factor which is directed along the ζ axis.

$$\ell_\zeta = F_\zeta / W \qquad (g) \qquad (2\text{-}62)$$

ℓ_T (Z(32)) The thrust load factor which is directed along the ξ axis.

$$\ell_T = T/W \qquad (g) \qquad (2\text{-}63)$$

ℓ_ξ (Z(33)) The total applied force load factor which is directed along the ξ axis.

$$\ell_\xi = \frac{T + F_A}{W} = \frac{F_\xi}{W} \qquad (g) \qquad (2\text{-}64)$$

ℓ_{AERO} (Z(34)) The total aerodynamic load factor.

$$\ell_{AERO} = \sqrt{\ell_\eta^2 + \ell_A^2 + \ell_\zeta^2} \qquad (g) \qquad (2\text{-}65)$$

$\ell_{APPLIED}$ (Z(35)) The total applied force load factor.

$$\ell_{APPLIED} = \sqrt{\ell_\eta^2 + \ell_\xi^2 + \ell_\zeta^2} \qquad (g) \qquad (2\text{-}66)$$

The following three parameters, F_β, F_V, and F_γ are obtained by transforming the F_ξ, F_η, and F_ζ forces from the Vehicle Coordinate System to the Relative Velocity Coordinate System by means of the coordinate transformation defined in Eq. (M-1). The algebraic sign is in accordance with the $\hat{i}_\beta$ - $\hat{i}_V$ - $\hat{i}_\gamma$ coordinate system which is defined in Section 2.2.2.1.

F_β	(Z(44))	The total applied force directed along the $\hat{i}_\beta$ axis. The resulting expression for F_β is presented in Eq. (2-42). (lb)
F_V	(Z(45))	The total applied force directed along the $\hat{i}_V$ axis. The resulting expression for F_β is presented in Eq. (2-42). (lb)
F_γ	(Z(46))	The total applied force directed along the $\hat{i}_\gamma$ axis. The resulting expression for F_β is presented in Eq. (2-42). (lb)

Attitude. The following three parameters (σ, α, λ) are defined according to their geometric meaning as given in Section 2.2.2 (Transformation from Vehicle Axes to Relative Velocity Axes) and in Section 5.1.2.1 (Attitude). These parameters and the ψ and $\dot{\psi}$ parameters which are listed subsequent to them are either evaluated directly by input or are computed according to the selected input options which are described in the sections under Section 4.2.3.6 (Attitude Models).

σ	(Z(11))	Roll (bank) angle which is defined in Sections 2.2.2.2 and 5.1.2.1. Except when using the bank angle modulation scheme, the current roll (bank) angle is evaluated directly from program input according to the selected roll angle control option. These options and the corresponding roll angle definitions are described in Section 4.2.3.6 (Roll (Bank) Angle Definition). For description of the bank angle modulation scheme for maintaining a constant time rate of change of relative flight path angle, see Section 2.2.2.6 (Specified Time Rate of Change of Relative Flight Path Angle). (deg)

α (Z(12)) Pitch angle of attack which is defined in Section 2.2.2.2 and 5.1.2.1. Except when using the pitch angle of attack modulation scheme or the pitch angle attitude control mode, the pitch angle of attack is evaluated directly from program input according to the selected pitch angle of attack control option. These options are described in Section 4.2.3.6 (Pitch Angle or Pitch Angle of Attack Definition). For description of the pitch angle of attack modulation scheme for maintaining a constant time rate of change of relative flight path angle, see Section 2.2.2.6 (Specified Time Rate of Change of Relative Flight Angle). If a pitch angle attitude definition mode is used, α is then determined from ψ and λ; σ is assumed to be zero when using the pitch angle attitude definition mode. From the definitions of α, λ, ψ, R, and V, the geometric relationships of these parameters can be illustrated as in Figure 2-11. α is then given by:

$$\left.\begin{aligned} \alpha &= 90° - \gamma - \psi_0 \\ \text{where} \quad \psi_0 &= \text{ARCCOS} \left(\frac{\cos\psi}{\cos\lambda}\right) \end{aligned}\right\} \quad \text{(deg)} \qquad (2\text{-}67)$$

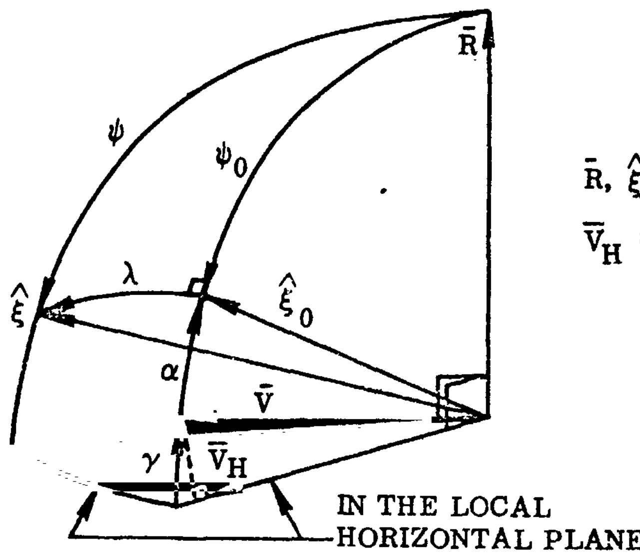
In the event that $|\cos\psi| > |\cos\lambda|$, the previous value of α is assumed. If $\lambda = 0$, then $\psi_0 = \psi$ and Eq. (2-67) becomes:

$$\alpha = 90° - \gamma - \psi \qquad \text{(deg)} \qquad (2\text{-}68)$$

λ (Z(13)) Yaw angle of attack which is defined in Sections 2.2.2.2 and 5.1.2.1. The yaw angle of attack is evaluated directly from program input according to the selected yaw angle of attack control option. These options are described in Section 4.2.3.6 (Yaw Angle of Attack Definition). If the

$\hat{\xi}_0$ is the unit vector projection of $\hat{\xi}$ in the $\bar{R}$ - $\bar{V}$ plane

$\bar{V}_H$ is the vector projection of $\bar{V}$ in the horizontal plane



$\bar{R}$, $\hat{\xi}_0$, $\bar{V}$, and $\bar{V}_H$ are coplanar

Figure 2-11. Pitch Angle of Attack Defined from the Pitch Angle

specified roll axis azimuth mode (e.g., "LAMC(K)" = 3., or 4.,) is used, λ is then determined from α or ψ, γ, β, and the specified roll axis azimuth (δ). For this mode, a zero roll angle (σ) is assumed. If it is necessary to initiate the simulation section with the value of δ at the end of the previous section or, in the case of Section 1, the value of δ which results from the initial value of λ, then this initial value, δ_0 is obtained by the following procedure (see Figure 2-12).

$$\dot{\psi}_0' = 90° - \alpha_0 - \gamma_0 \qquad \text{(deg)} \qquad (2\text{-}69)$$

where γ_0 is the flight path angle at the initiation of the current simulation section. If $\dot{\psi}$ is within 0.0000001° of either 0° or 180°, δ_0 is assumed to be zero. If λ_0 is within 0.0000001° of ±90°, then the horizontal component of λ_0 (λ_{H_0}) is equal to λ_0, otherwise:

$$\lambda_{H_0} = \mathrm{ARCTAN}\left(\frac{\tan \lambda_0}{\sin \psi_0'}\right) \qquad \text{(deg)} \qquad (2\text{-}70)$$

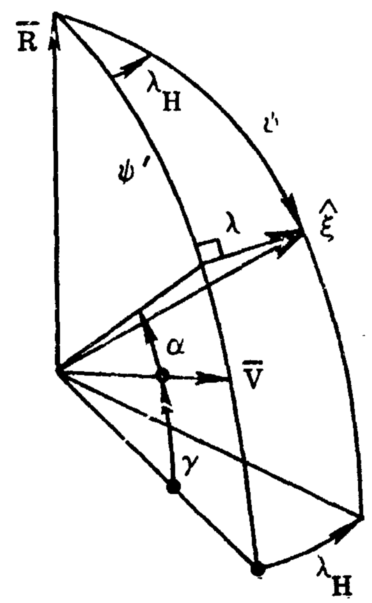


Figure 2-12. Geometry for the Specified Roll Axis Azimuth Mode

δ_0 is then expressed by

$$\delta_0 = \beta_0 + \lambda_{H_0}$$

When α is specified, δ at the section relative time t_R is determined according to the following scheme (see Figure 2-12.

$$\psi' = 90° - \alpha - \gamma \quad \text{(deg)} \qquad (2\text{-}71)$$

$$\lambda_H = \dot{\delta}\, t_R + \delta_K + \delta_0 - \beta \quad \text{(deg)} \qquad (2\text{-}72)$$

where δ_K is a constant input angular term.

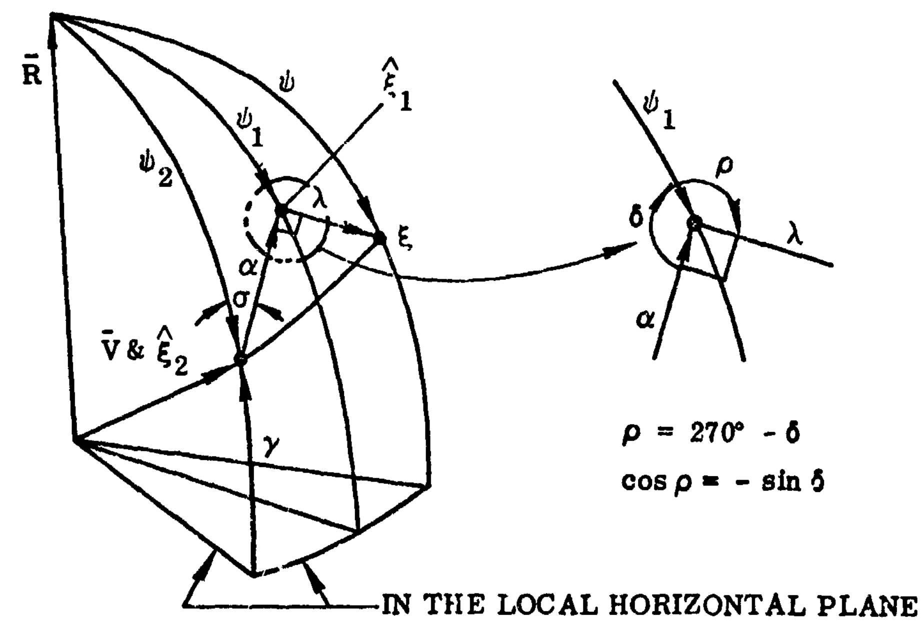
If ψ' is within 0.0000001° of either 0° or 180° or if the value of λ_H is within 0.0000001° of ±90°, then the value of λ is assumed to be equal to that of λ_H, otherwise:

$$\lambda = \text{ARCTAN}\,(\sin \psi' \tan \lambda_H) \quad \text{(deg)} \qquad (2\text{-}73)$$

For the case where ψ, rather than α, is specified, δ at the section relative time t_R is determined according to the following procedure. λ_H is determined by Eq. (2-72); then if λ_H is within 0.0000001° of 0°, ±90°, or 180°, the value of λ is equal to that of λ_H, otherwise:

$$\lambda = \text{ARCSIN} (\sin \lambda_H \sin \psi) \qquad \text{(deg)} \qquad (2\text{-}74)$$

ψ (Z(43)) Relative pitch angle which is the angle between the current geocentric radius vector (up direction) to the vehicle roll axis (ξ axis). This angle is either evaluated directly from program input according to the selected pitch angle control option (these options are described in Section 4.2.3.6. Pitch Angle of Pitch Angle of Attack Definition) or as a teometric relationships of these parameters are illustrated in Figure 2-13.



NOTE: $\hat{\xi}$ and $\hat{\xi}_1$ vectors have been omitted to keep the sketch simple.

Figure 2-13. Pitch Angle Defined from the Pitch Angle of Attack

For general values of these parameters, ψ is obtained by the following procedure:

$$\left.\begin{aligned}
\psi_2 &= 90^\circ - \gamma \\
\psi_1 &= \text{ARCCOS}\,(\cos\psi_2 \cos\alpha + \sin\psi_2 \sin\alpha\cos\sigma) \\
\delta &= \text{ARCCOS}\left(\frac{\cos\psi_2 - \cos\psi_1 \cos\alpha}{\sin\psi_1 \sin\alpha}\right) \\
\psi &= \text{ARCCOS}\,(\cos\psi_1 \cos\lambda - \sin\psi_1 \sin\lambda \sin\delta)
\end{aligned}\right\}\text{(deg)} \qquad (2\text{-}75)$$

The above equations can be simplified if σ and/or α and/or λ are equal to zero. If $\sigma = 0$, then:

$$\left.\begin{aligned}
\psi_1 &= 90^\circ - \gamma - \alpha \text{ and } \delta = 180^\circ \\
\psi &= \text{ARCCOS}\,(\cos\psi_1 \cos\lambda)
\end{aligned}\right\}\text{(deg)} \qquad (2\text{-}76)$$

If $\lambda = 0$, then $\psi = \psi_1$ and, consequently, the expressions for δ and ψ listed in Eq. (2-75) need not be evaluated. If both σ and λ equal zero, then:

$$\psi = 90^\circ - \gamma - \alpha \qquad \text{(deg)} \qquad (2\text{-}77)$$

Logic has been provided in the program to use these simplifications when appropriate in order to reduce the number of trigonometric functions which must be evaluated and the number of required arithmetic operations.

$\dot{\psi}$ (Z(78)) The relative pitch rate which is the time rate of change of the relative pitch angle (ψ). The algebraic sign is positive for increasing values of the pitch angle. This rate is either specified directly by program input according to the selected pitch angle control option (these options are described in Section 4.2.3.6, Pitch Angle or Pitch Angle of Attack Definition) or as a function of the ψ time history. In this latter case, $\dot{\psi}$ is expressed as:

$$\dot{\psi} = \frac{\psi - \psi_0}{\Delta t} \qquad \text{(deg/sec)} \qquad (2\text{-}78)$$

where

ψ_0 is the value of ψ at the completion of the previously successful integration step

Δt is the time from the completion of the previously successful integration step

<u>Environment</u>

q (Z(9)) The dynamic pressure (with p in lb/in.^{2}):

$$q = 100.8\, p\, M^2 \qquad (\text{lb/ft}^2) \qquad (2\text{-}79)$$

$\log_{10} q$ (Z(92)) The logarithm to the base 10 of q. $\text{Log}_{10}\, q$ is defined according to:

$$\text{If} \quad q > 0.001 \text{ lb/ft}^2 \quad \text{ARG} = q \qquad (2\text{-}80)$$

$$\text{If} \quad q \leq 0.001$$

$$\text{ARG} = \frac{9.6 \left(\frac{V+1}{25800}\right)^2}{10^{\left(\frac{|h - 260000|}{51000}\right)}} \qquad (2\text{-}81)$$

$$\text{Then} \quad \log_{10} q = \log_{10} (\text{ARG}) \qquad (2\text{-}82)$$

$\bar{g}$ (Z(10)) The total gravitational acceleration vector. $\bar{g}$ is obtained directly from the central body gravitational potential energy function U by taking its gradient (∇U). The expression for U, obtained from Reference 5, is:

$$U = -\frac{\mu}{R}\left[1 - \sum_{n=2}^{N} J_n \left(\frac{A}{R}\right)^n P_n\right] \qquad (\mathrm{ft}^2/\mathrm{sec}^2) \qquad (2\text{-}83)$$

where

A is the equatorial radius

J_n is the n-th gravitational potential harmonic coefficient

N is the highest order harmonic which is considered

P_n is the n-th Legendre polynomial in $\sin\theta$ terms

μ is the gravitational field constant of the central body

The gravity model which is used in GTSM includes up through the fourth harmonic term ($n = 2, 3, 4$) and assumes symmetry about the rotational axis. The corresponding Legendre polynomials in $\sin\theta$ terms are:

$$\left.\begin{aligned} P_2 &= 1/2\,(3\sin^2\theta - 1) \\ P_3 &= 1/2\,(5\sin^3\theta - 3\sin\theta) \\ P_4 &= 1/8\,(35\sin^4\theta - 30\sin^2\theta + 3) \end{aligned}\right\} \qquad (2\text{-}84)$$

The gravitational acceleration vector $\bar{g}$ is then given by:

$$\bar{g} = \nabla U = -\cancelto{0}{\left(\frac{1}{R\sin\theta}\right)\left(\frac{\partial U}{\partial \phi}\right)\hat{i}_\phi} - \left(\frac{1}{R}\right)\left(\frac{\partial U}{\partial \theta}\right)\hat{i}_\theta - \left(\frac{\partial \mu}{\partial R}\right)\hat{i}_R \qquad (2\text{-}85)$$

$$(\mathrm{ft}/\mathrm{sec}^2)$$

Since symmetry is assumed about the rotational axis (e_3 axis), the first right hand term of Eq. (2-85) is zero. By convention the component magnitudes of $\bar{g}$ are defined:

$$\left.\begin{aligned} g_\theta &= -\left(\frac{1}{R}\right)\left(\frac{\partial U}{\partial \theta}\right) \\ g_R &= +\left(\frac{\partial U}{\partial R}\right) \end{aligned}\right\} (\mathrm{ft}/\mathrm{sec}^2) \qquad (2\text{-}86)$$

Then $\bar{g}$ can be written:

$$\left.\begin{aligned} \bar{g} &= \bar{g}_\theta + \bar{g}_R = g_\theta \hat{i}_\theta - g_R \hat{i}_R \\ g &= \sqrt{g_\theta^{\ 2} + g_R^{\ 2}} \end{aligned}\right\} \text{(ft/sec}^2\text{)} \qquad (2\text{-}87)$$

R_s (Z(14)) The local surface radius of the central body. The surface in general, can be defined as an oblate spheroid (see Figure 2-14) with symmetry about the rotational axis and the equatorial plane. This surface is described mathematically as a symmetric ellipsoid. The local surface radius is then expressed as a function of the geocentric latitude (θ) from the equation which defines the elliptic cross section containing the rotational axis.

$$R_s = \left[\left(\frac{\cos\theta}{A}\right)^2 + \left(\frac{\sin\theta}{B}\right)^2\right]^{-1/2} \quad \text{(ft)} \qquad (2\text{-}88)$$

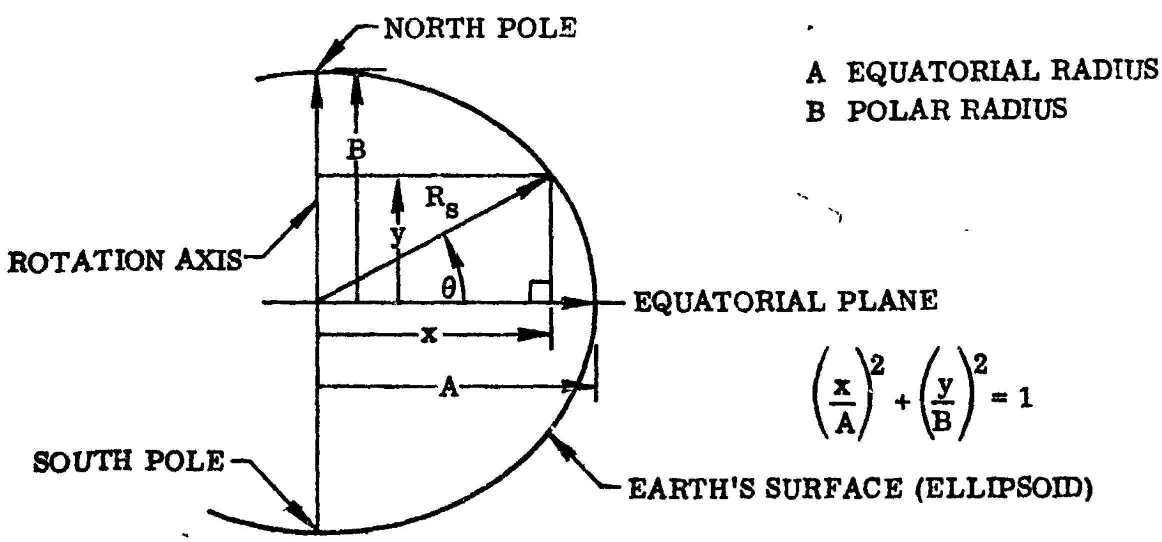


Figure 2-14. Central Body Surface Geometry

$\bar{g}_R$ (Z(15)) The local gravitational acceleration along the geocentric radius vector. The algebraic sign is positive for the downward direction toward the central body. $\bar{g}_R$ is readily obtained from Eq. (2-86), (2-83), and (2-84).

$$\left.\begin{aligned} g_R &= \left(\frac{\partial U}{\partial R}\right) = \frac{\mu}{R^2}\left[1 - \sum_{n=2}^{4} (n+1)\, J_n P_n \left(\frac{A}{R}\right)^n\right] \\ &\qquad\qquad\qquad\qquad\qquad (\text{ft/sec}^2) \\ \bar{g}_R &= -\, g_R\, \hat{i}_R \end{aligned}\right\} \quad (2\text{-}89)$$

$\bar{g}_\theta$ (Z(16)) The local gravitational acceleration along the $\hat{i}_\theta$ axis with positive direction toward the north. $\bar{g}_\theta$ is readily obtained from Eq. (2-86), (2-83), and (2-84).

$$\left.\begin{aligned} g_\theta &= -\left(\frac{1}{R}\right)\left(\frac{\partial U}{\partial \theta}\right) = -\frac{\mu}{R^2} \sum_{n=2}^{4} J_n \left(\frac{A}{R}\right)^n \frac{d P_n}{d\theta} \\ &\qquad\qquad\qquad\qquad\qquad (\text{ft/sec}^2) \\ \bar{g}_\theta &= g_\theta\, \hat{i}_\theta \end{aligned}\right\} \quad (2\text{-}90)$$

where

$$\frac{dP_2}{d\theta} = 3 \cos\theta \sin\theta$$

$$\frac{dP_3}{d\theta} = \frac{3}{2}\left[5 \sin^2\theta - 1\right] \cos\theta$$

$$\frac{dP_4}{d\theta} = \frac{5}{2}\left[7 \sin^3\theta - 3 \sin\theta\right] \cos\theta$$

p (Z(36)) The current atmospheric ambient pressure which is assumed to be a function of altitude. p is obtained directly from program input. See Section 4.2.3.6 (Atmosphere Model). (lb/in.^{2})

V_s (Z(37)) The current velocity of sound which is assumed to be a function of altitude. V_s is obtained directly from program input. See Section 4.2.3.6 (Atmosphere Model). (ft/sec)

H_F (Z(38)) The current heat flux which is the product of the current dynamic pressure and the relative velocity.

$$H_F = q\,V \qquad (\text{lb}/[\text{ft-sec}]) \qquad (2\text{-}91)$$

M (Z(40)) The current Mach number

$$M = \frac{V}{V_s} \qquad (2\text{-}92)$$

<u>Aerodynamics</u>

C_η (Z(21)) The aerodynamic yaw force coefficient* along the standard pitch axis ($\hat{\eta}$ axis) with algebraic sign in accordance with the Vehicle Coordinate System described in Section 2.2 ?.1 (Positive to the Right). C_η is assumed to be a function of Mach number and yaw angle of attack (λ). Symmetry is assumed about $\lambda = 0$. The value of C_η is obtained directly from program input. See Section 4.2.3.6, Aerodynamic Yaw Force Coefficient Table (C_η Table).

C_ξ (Z(22)) The aerodynamic axial force coefficient along the standard roll axis ($\hat{\xi}$ axis) with algebraic sign in accordance with the Vehicle Coordinate System described in Section 2.2.2.1 (Positive Forward). C_ξ is assumed to be a function of Mach number and pitch angle of attack. The value of C_ξ is obtained directly from program input. An option has

* This parameter is suppressed (i.e., set equal to zero) in the current version of the Space Shuttle Synthesis Program. The skeleton logic has been retained, however, to simplify the incorporation of this parameter if this should become desirable.

been provided in which two separately obtained (i.e., from separate tables) C_ξ coefficients are combined algebraically to form an effective C_ξ value. This option is used when modeling both booster and orbiter vehicles simultaneously during booster burn. See Section 4.2.3.6, Aerodynamic Axial Force Coefficient Table (C_ξ Table).

C_ζ (Z(23)) The aerodynamic normal force coefficient along the standard yaw axis ($\hat{\zeta}$ axis) with algebraic sign in accordance with the Vehicle Coordinate System described in Sedtion 2.2.2.1 (Positive Downward). C_ζ is assumed to be a function of Mach number and pitch angle of attack. The value of C_ζ is obtained directly from program input. An option has been provided in which two separately obtained (i.e., from separate tables) C_ζ coefficients are combined algebraically to form an effective C_ζ value. This option is used when modeling both booster and orbiter vehicles simultaneously during booster burn. See Section 4.2.3.6 Aerodynamic Normal Force Coefficient Table (C_ζ Table).

<u>Weight</u>

W (Z(7)) The current total weight of the vehicle. This weight is the result of integration of $\dot{W}$ (see Section 2.2.2.4 Weight) and the initializing and/or the jettisoning of weight at discrete points in the trajectory (see "JETW(K)" listed in Section 4.2.3.6 Weight Models). (lb)

<u>Propulsion</u>. All propulsion models currently within GTSM are SIMPO models. A SIMPO model is a propulsion model defined by three propulsion parameters from which the current propulsive thrust (T), propellant flow rate ($\dot{W}_{PROP}$), and Specific

Impulse (I_{sp}) can be obtained. GTSM has nine* SIMPO model options per simulation section and two propulsion table options per propulsion table (see Section 4.2.3.6 Propulsion Models). The propulsion tables can be considered as SIMPO models in which one or two of the necessary propulsion parameters, depending on which table option is being used, is (are) defined at the current time via table look-up. By including the table options and the zero thrust case, there are eleven basic procedures by which T, $\dot{W}_{PROP}$, and I_{sp} can be defined.

a. Zero thrust case; T and I_{sp} are assumed to be zero and $\dot{W}_{PROP}$ is input directly.

b. Given the vacuum thrust (T_{VAC}) in lb), the sea level thrust (T_{SL} in lb), and the propellant flow rate ($\dot{W}$ in lb/sec), then the current thrust (T) and Specific Impulse (I_{sp}) are obtained by assuming that W is constant and that T and I_{sp} are uniquely defined according to a linear independence on atmospheric pressure (p).

$$\left.\begin{aligned} T &= T_{VAC} - (T_{VAC} - T_{SL})(p/p_0) && \text{(lb)} \\ I_{sp} &= \frac{T}{\dot{W}} && \text{(sec)} \end{aligned}\right\} \qquad (2\text{-}93)$$

where p_0 is the sea level atmospheric pressure.

c. Given the vacuum thrust (T_{VAC} in lb), the vacuum specific impulse (I_{VAC} in sec), and the sea level specific impulse (I_{SL} in sec), then the current specific impulse (I_{sp}), propellant flow rate ($\dot{W}$), and thrust level (T) are obtained by assuming that W is constant and that T and I_{sp} are uniquely defined by the atmospheric pressure (p).

---

* Only two options are used during the ascent trajectory. These options, which are internally set are the option described in paragraphs a and g. All options are available during the return trajectory.

$$
\left.\begin{aligned}
I_{sp} &= I_{VAC} - (I_{VAC} - I_{SL})(p/p_0) && \text{(sec)} \\
\dot{W} &= T_{VAC}/I_{VAC} && \text{(lb/sec)} \\
T &= I_{sp}\,\dot{W} && \text{(lb)}
\end{aligned}\right\} \quad (2\text{-}94)
$$

d. Given a constant thrust level (T_c in lb), the vacuum specific impulse (I_{VAC} in sec), and the sea level specific impulse (I_{SL} in sec), then the current specific impulse (I_{sp}), and the current propellant flow ($\dot{W}$) are defined uniquely by assuming a linear dependence on atmospheric pressure. This option describes a propulsion system in which the propellant flow rate is modulated to maintain a constant thrust.

$$
\left.\begin{aligned}
T &= T_c && \text{(lb)} \\
I_{sp} &= I_{VAC} - (I_{VAC} - I_{SL})\,(p/p_0) && \text{(sec)} \\
\dot{W} &= T/I_{sp} && \text{(lb/sec)}
\end{aligned}\right\} \quad (2\text{-}95)
$$

e. Given the vacuum thrust (T_{VAC} in lb), the propellant flow rate ($\dot{W}$ in lb/sec), and the total engine exit area (A_{EXIT} in in.^{2}), then the current thrust level (T) and specific impulse (I_{sp}) are defined as a linear function of atmospheric pressure (p), and $\dot{W}$ is assumed to be constant.

$$
\left.\begin{aligned}
T &= T_{VAC} - p\,A_{EXIT} && \text{(lb)} \\
I_{sp} &= T/\dot{W} && \text{(sec)}
\end{aligned}\right\} \quad (2\text{-}96)
$$

f. Given the sea level thrust (T_{SL} in lb), the propellant flow rate ($\dot{W}$ in lb/sec), and the total engine exit area (A_{EXIT} in in.^{2}), then the current thrust level (T) and specific impulse (I_{sp}) are defined as a linear function of atmospheric (p), and $\dot{W}$ is assumed to be constant.

$$\left.\begin{aligned} T &= T_{SL} + (p_0 - p)\, A_{EXIT} && \text{(lb)} \\ I_{sp} &= T/\dot{W} && \text{(sec)} \end{aligned}\right\} \quad (2\text{-}97)$$

g. Given the maximum vacuum thrust ($[T_{VAC}]_{MAX}$ in lb), the vacuum specific impulse (I_{VAC} in sec), the sea level specific impulse (I_{SL} in sec), and an engine throttling condition which is the maximum allowable total applied force load factor along the $\hat{\xi}$ axis ($[\ell_\xi]_{MAX}$ in g's), then for $\ell_\xi \le [\ell_\xi]_{MAX}$ the current specific impulse (I_{sp}), propellant flow rate ($\dot{W}$), and thrust level (T) are obtained by assuming that $\dot{W}$ is constant and that T and I_{sp} are uniquely determined by the atmopsheric pressure (p) by Eq. (2-94). When $\ell_\xi > [\ell_\xi]_{MAX}$ (the throttling condition), the thrust level is defined such that the resulting $\ell_\xi = [\ell_\xi]_{MAX}$. It is noted that Eq. (2-94) is solved prior to testing the value of ℓ_ξ in order to obtain the I_{sp} value.

$$\left.\begin{aligned} T &= W\, [\ell_\xi]_{MAX} - F_A && \text{(lb)} \\ \dot{W} &= T/I_{sp} && \text{(lb/sec)} \end{aligned}\right\} \quad (2\text{-}98)$$

The negative sign in the thrust equation results from the sign convention used for the aerodynamic forces.

h. This option is identical to that described in g above, except that the maximum sea level thrust ($[T_{SL}]_{MAX}$ in lb) is specified instead of $[T_{VAC}]_{MAX}$. In this case the second equation of Eq. (2-94) is replaced with:

$$\dot{W} = T_{SL}/I_{SL} \qquad \text{(lb/sec)} \qquad (2\text{-}99)$$

All other computations are the same as those defined in g above.

i. This option is identical to that described in g above, except that the maximum allowable thrust load factor ($[\ell_T]_{MAX}$ in g's) is specified rather than $[\ell_\xi]_{MAX}$. In this case, the throttling condition is based on the value of ℓ_T and the first equation of Eq. (2-98) is replaced with:

$$T = W\,[\ell_T]_{MAX} \qquad \text{(lb)} \qquad (2\text{-}100)$$

j. This option is identical with that described in i above, except that the maximum sea level thrust ($[T_{SL}]_{MAX}$ in lb) is specified instead of $[T_{VAC}]_{MAX}$. In this case, the second equation of Eq. (2-94) is replaced with Eq. (2-99).

k. Given the vacuum thrust (T_{VAC} in lb), the effective vacuum specific impulse (I_{VAC} in sec), and the engine exit area (A_{EXIT} in in.^{2}), then the current propellant flow rate ($\dot{W}$), thrust level (T), and specific impulse (I_{sp}) are determined as a function of atmospheric pressure (p):

$$\left.\begin{aligned}
\dot{W} &= N_{ENG}\left(\frac{T_{VAC}}{I_{VAC}}\right) && \text{(lb/sec)}\\
T &= N_{ENG}\,(T_{VAC} - pA_{EXIT})\cos\Delta && \text{(lb)}\\
I_{sp} &= \frac{T}{\dot{W}} && \text{(sec)}
\end{aligned}\right\} \quad (2\text{-}101)$$

where:

N_{ENG} is the number of motor groups simulated by the input value of T_{VAC}, e.g., two solid strapons with motor data input for one solid.

Δ is the motor cant angle. Propulsive components due to a cant angle which are not parallel to the $\hat{\xi}$ axis are assumed to cancel out by symmetric placement of the motors each one of which are assumed to have the same cant angle.

This procedure is used for the propulsion table option in which T_{VAC} is specified as a function of a time argument and a constant effective vacuum specific impulse is assumed. When both T_{VAC} and $\dot{W}$ are specified as a function of a time argument, the same procedure is used except that $\dot{W}$ is the product of N_{ENG} and the $\dot{W}$ which is obtained from the $\dot{W}$ table.

Logic has been provided in the program to handle special cases for those options in which no difference between sea level and vacuum conditions is specified.

2.2.2.9 <u>After-The-Fact Parameters</u>. The after-the-fact trajectory parameters are those parameters which do not need to be evaluated in order to solve the twelve differential equations. This is significant in that it is not necessary to evaluate these parameters until after an integration step has been successfully completed, consequently, these parameters are not computed as often as the so-called necessary trajectory parameters.

<u>Position</u>

DR (Z(49)) The downrange angle. DR is the central angle between the initial vector and the current position vector (see Figure 4-3 in Section 4.2.3.3 (Targeting). The initial vector ($\hat{L}$) is a unit geocentric vector with direction defined by the initial latitude (θ_L) and the initial longitude (ϕ_L). The unit position vector ($\hat{R}$) is a unit geocentric vector with direction defined by the current latitude (θ) and the current longitude (ϕ). $\hat{L}$ and $\hat{R}$ are expressed in the Earth-Fixed Coordinate system as:

$$\left.\begin{aligned} \hat{L} &= (\cos\theta_L \cos\phi_L)\,\hat{e}_1 + (\cos\theta_L \sin\phi_L)\,\hat{e}_2 + (\sin\theta_L)\,\hat{e}_3 \\ \hat{R} &= (\cos\theta \cos\phi)\,\hat{e}_1 + (\cos\theta \sin\phi)\,\hat{e}_2 + (\sin\theta)\,\hat{e}_3 \end{aligned}\right\} \quad (2\text{-}102)$$

Define an angular argument ARG_{DR} such that:

$$ARG_{DR} = ARCCOS\,(\hat{L} \cdot \hat{R})$$

Then from the geometry

$$\left.\begin{array}{l} \text{if } (\beta_0 - 180°)\,([\hat{L} \times \hat{R}] \cdot \hat{e}_3) > 0,\ DR = 360° - ARG_{DR} \\ \text{if } (\beta_0 - 180°)\,([\hat{L} \times \hat{R}] \cdot \hat{e}_3) \le 0,\ DR = ARG_{DR} \qquad \text{(deg)} \end{array}\right\} \quad (2\text{-}103)$$

where

β_0 is the initial relative azimuth at the initiation of the trajectory simulation and $[\hat{L} \times \hat{R}] \cdot \hat{e}_3$ defines the component of $\hat{L} \times \hat{R}$ along the $\hat{e}_3$ axis. The algebraic sign is positive if $\hat{L} \times \hat{R}$ lies in the northern hemisphere.

CR (Z(50)) The crossrange angle. CR is the angle between two planes S_1 and S_2 where S_1 is the plane which contains the initial vector $(\hat{L})$ and a reference target vector $(\hat{T})$, S_2 is the plane which contains the initial vector $(\hat{L})$ and the unit position vector $(\hat{R})$. See Figure 4-3 in Section 4.2.3.3 (Targeting). The reference target vector $(\hat{T})$ is a unit geocentric vector with direction defined by the target latitude (θ_T) and the target longitude (ϕ_T). $\hat{T}$ is also expressed in the Earth-Fixed Coordinate system as:

$$\hat{T} = (\cos\theta_T \cos\phi_T)\,\hat{e}_1 + (\cos\theta_T \sin\phi_T)\,\hat{e}_2 + (\sin\theta_T)\,\hat{e}_3 \quad (2\text{-}104)$$

The unit vector in the western direction $(\hat{W})$ at the initial coordinates is expressed

$$\hat{W} = (\sin\phi_L)\,\hat{e}_1 - (\cos\phi_L)\,\hat{e}_2 \quad (2\text{-}105)$$

Define an angular argument ARG_{β_R} such that:

$$ARG_{\beta_R} = ARCCOS\left(\frac{(\hat{L} \times \hat{R}) \cdot \hat{W}}{|\hat{L} \times \hat{R}|}\right) \quad \text{(deg)} \quad (2\text{-}106)$$

Then from geometry, the relative azimuth of plane S_2 at the initial coordinates θ_L and ϕ_L is given by

$$\left.\begin{array}{l} \text{if } [\hat{L} \times \hat{R}] \cdot \hat{e}_3 \geq 0, \ \beta_R = ARG_{\beta_R} \\ \text{if } [\hat{L} \times \hat{R}] \cdot \hat{e}_3 < 0, \ \beta_R = 360° - ARG_{\beta_R} \end{array}\right\} \text{(deg)} \qquad (2\text{-}107)$$

The relative azimuth of plane S_1 at the initial coordinates θ_L and ϕ_L is given by

$$\beta_T = ARCCOS\left(\frac{(\hat{L} \times \hat{T}) \cdot \hat{W}}{|\hat{L} \times \hat{T}|}\right) \qquad \text{(deg)} \qquad (2\text{-}108)$$

Then crossrange becomes

$$CR = ARG_{\beta_R} - \beta_T \qquad \text{(deg)} \qquad (2\text{-}109)$$

It is noted that there are more direct vector methods which could be used to obtain CR. These methods require taking either the dot or the cross product of $(\hat{L} \times \hat{R})$ and $(\hat{L} \times \hat{T})$. However, crossrange is used frequently when iterating to specified coordinates (θ_T, ϕ_T) and, consequently, values of CR near zero must be accurately determined. For this reason it is not desirable to dot multiply $(\hat{L} \times \hat{R})$ and $(\hat{L} \times \hat{T})$ since this results in defining the cosine of CR which does not have acceptable resolution for small values of CR. The cross product of $(\hat{L} \times \hat{R})$ and $(\hat{L} \times \hat{T})$ defines a vector which in absolute value can be used to define the sine of CR. The problem here is that the algebraic sign is lost when taking the absolute value of the vectors, hence, additional logic would be needed.

T_{MISS} (Z(51)) The target miss angle, T_{MISS} is the central angle between the unit position vector $(\hat{R})$ and the reference target vector $(\hat{T})$. Since there is no algebraic sign associated with this distance angle, T_{MISS} can be determined directly from the cross product of $\hat{R}$ and $\hat{T}$.

$$T_{MISS} = \text{ARCSINE} \left(\frac{|\hat{R} \times \hat{T}|}{|\hat{R}|\,|\hat{T}|} \right) \quad \text{(deg)} \qquad (2\text{-}110)$$

β_R (Z(52)) The relative azimuth of plane S_2 at the initial coordinates θ_L and ϕ_L is defined by Eq. (2-106) and (2-107).

DR_T (WQ(7,M)) The target down range central angle is determined by the same procedure as DR except the vector $\hat{T}$ replaces the vector $\hat{R}$ and, correspondingly, θ_T and ϕ_T replace θ and ϕ, respectively.

β_T (WQ(8,M)) (WQ(9,M)) The relative azimuth of plane S_1 at the initial coordinates θ_L and ϕ_L where β_T is expressed by Eq. (2-108) and $0° \leq$ WQ(8,M) < 180°. Logic similar to that of Eq. 2-107 (except that the vector $\hat{T}$ replaces the vector $\hat{R}$) is applied when determining WQ(9,M), consequently $0° \leq$ WQ(9,M) <360°.

Velocity

V_I (Z(41)) The inertial velocity magnitude (V_I). V_I is defined by expressing both the inertial velocity vector $(\bar{V}_I)$ and the relative velocity vector (V) in the Local Coordinate System. The velocity vector $(\bar{V}_\oplus)$ due to the rotation of the central body is:

$$\bar{V}_\oplus = \omega R \hat{i}_\phi \qquad (2\text{-}111)$$

Let V_{I_ϕ}, V_{I_θ}, and V_{I_R} denote the components of V_I along the i_ϕ, i_θ, and i_R axes, respectively (see Figure 2-15). Then by geometry and Eq. (M-5).

$$\left.\begin{aligned} V_{I_\phi} &= \omega R \cos\theta + V(\sin\beta\cos\gamma) \\ V_{I_\theta} &= V(\cos\beta\cos\gamma) \\ V_{I_R} &= V(\sin\gamma) \end{aligned}\right\} \text{(ft/sec)} \qquad \text{(2-112)}$$

By root-sum-squaring:

$$V_I = \sqrt{V_{I_\phi}^2 + V_{I_\theta}^2 + V_{I_R}^2} \qquad \text{(ft/sec)} \qquad \text{(2-113)}$$

γ_I (Z(42)) The inertial flight path angle (γ_I). γ_I is the angle between $\bar{V}_I$ and $\bar{V}_{I_H}$ and is defined by

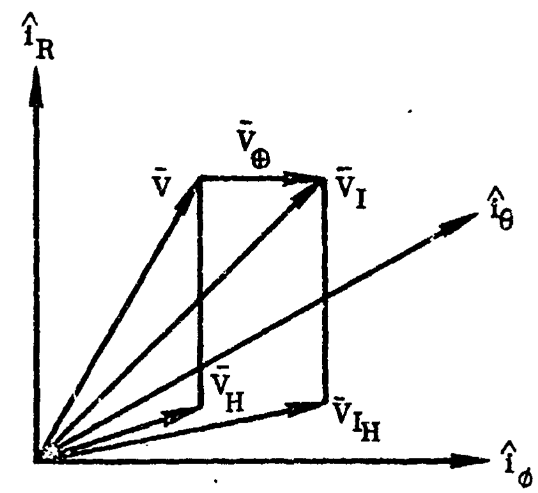
$$\gamma_I = \text{ARCSIN}\left(\frac{V_{I_R}}{V_I}\right) \qquad \text{(deg)} \qquad \text{(2-114)}$$

β_I (Z(65)) The inertial azimuth (β_I). β_I is the angle between $\hat{i}_\theta$ and $\bar{V}_{I_H}$ and is expressed

$$\text{ARG}_{\beta_I} = \text{ARCCOS}\left(\frac{V_{I_\theta}}{V_I \cos\gamma_I}\right) \qquad \text{(deg)} \qquad \text{(2-115)}$$

$$\left.\begin{aligned} &\text{if } V_{I_\phi} \geq 0, \text{ then } \beta_I = \text{ARG}_{\beta_I} \\ &\text{if } V_{I_\phi} < 0, \text{ then } \beta_I = 360° - \text{ARG}_{\beta_I} \end{aligned}\right\} \text{(deg)} \qquad \text{(2-116)}$$

Figure 2-15 illustrates the above relationships in diagrammatic form.



$\bar{V}_{I_H}$ is the projection of the inertial velocity vector ($\bar{V}_I$) in the horizontal plane which is defined by $\hat{i}_\theta$ and $\hat{i}_\phi$

$\bar{V}_H$ is the projection of $\bar{V}_R$ in the horizontal plane

$\bar{V}_\oplus$ is the velocity vector due to the rotation of the central body

β_I is the inertial azimuth, the angle between $\hat{i}_\theta$ and $\bar{V}_{I_H}$

β is the relative azimuth, the angle between $\hat{i}_\theta$ and $\bar{V}_H$

γ_I is the inertial flight path angle, the angle between $\bar{V}_I$ and $\bar{V}_{I_H}$

γ is the relative flight path angle, the angle between $\bar{V}$ and $\bar{V}_H$

Figure 2-15. Inertial Velocity – Relative Velocity Vector Relationship

2.2.2.10 Optional After-The-Fact Parameters. The optional after-the-fact parameters are those parameters which are computed only if specifically called for by program input. These parameters currently consist of the orbital elements (includes the instantaneous impact parameters), the inertial/relative components $\bar{R}$, $\bar{V}$, $\bar{V}_I$, and $\bar{a}$, and the radar coordinate parameters.

Orbital Elements

i (Z(66)) Orbital inclination. The orbital inclination (i) can be defined directly from the geocentric latitude (θ) and the inertial azimuth (β_I) by solving a spherical triangle. See Figure 2-16.

$$i = \text{ARCCOS}(\cos\theta \sin\beta_I) \qquad \text{(deg)} \qquad (2\text{-}117)$$

Appropriate logic is provided in the program to recognize the condition of either zero V_I or $\gamma_I = \pm 90$ degree. Under these circumstances both i and the nodal longitude are set to zero.

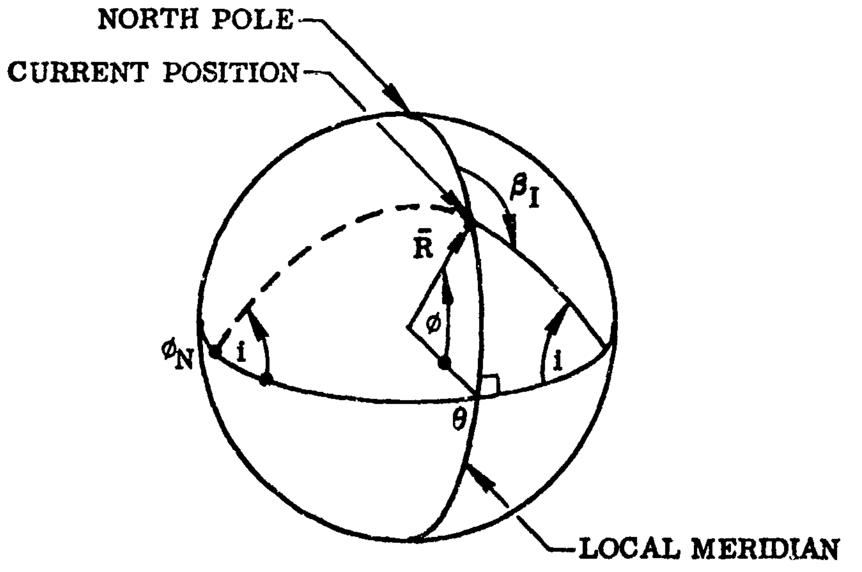


Figure 2-16. Orbital Inclination and Nodal Longitude

ϕ_N (Z(67)) Longitude of the ascending node (ϕ_N). The longitude of the ascending node (nodal longitude) is the longitude defined by the intersection of the current orbital plane and the equatorial plane at the node (point of intersection on the reference sphere) at which the position passes from the southern to the northern hemisphere. With reference to the geometry presented in Figure 2-16, an argument of longitude (ARG_{LNG}) is defined:

$$ARG_{LNG} = ARCCOSINE \left(\frac{\cos \beta_I}{\sin i} \right) \quad \text{(deg)} \qquad (2\text{-}118)$$

If the inclination is either zero or 180 degrees, the orbital plane is coplanar with the equatorial plane and, consequently, the nodal longitude is undefined. In this case the nodal longitude is assumed to be the current longitude (ϕ). If i is 90 degrees, the nodal longitude is equivalent to the current longitude (ϕ) if $\beta_I = 0$ degrees; if $\beta_I = 180$ degrees then ϕ_N is set equal to $\phi + 180$ degrees.* For the normally defined cases in which i does not equal either 0, 90, or 180 degrees, the nodal longitude is defined in the following table.

Table 2-3. Nodal Longitude Definition

Position and Velocity Conditions	Nodal Longitude*
$\theta > 0,\ 0° < \beta_I \le 90°,\ i < 90°$	$\phi_N = \phi - ARG_{LNG}$
$\theta > 0,\ 180° > \beta_I > 90°,\ i < 90°$	$\phi_N = \phi - ARG_{LNG}$
$\theta < 0,\ 0° < \beta_I \le 90°,\ i < 90°$	$\phi_N = \phi + ARG_{LNG}$
$\theta < 0,\ 180° > \beta_I > 90°,\ i < 90°$	$\phi_N = \phi + ARG_{LNG}$
$\theta > 0,\ 180° < \beta_I \le 270°,\ i > 90°$	$\phi_N = \phi + ARG_{LNG}$
$\theta > 0,\ 360° > \beta_I > 270°,\ i > 90°$	$\phi_N = \phi + ARG_{LNG}$
$\theta < 0,\ 180° < \beta_I \le 270°,\ i > 90°$	$\phi_N = \phi - ARG_{LNG}$
$\theta < 0,\ 360° > \beta_I > 270°,\ i > 90°$	$\phi_N = \phi - ARG_{LNG}$

(2-119) (deg)

djusted, if necessary, to yield $0 \le \phi_N < 360°$

C_3 (Z(68)) Orbital energy (C_3) in English units.

$$C_3 = V_I^2 - \frac{2\mu}{R} \qquad (ft^2/sec^2) \qquad (2\text{-}120)$$

where μ is the gravitational field constant of the central boc expressed in ft^3/sec^2.

V_c (Z(69)) Local circular velocity (V_c)

$$V_c = \sqrt{\frac{\mu}{R}} \qquad (ft/sec) \qquad (2\text{-}121)$$

R_A (Z(70)) Current geocentric apogee radius magnitude (R_A). R_A is only defined for elliptic orbits; in the event that parabolic or hyperbolic conditions exist, R_A is set equal to zero. For elliptic orbits, R_A is expressed as:

$$R_A = \frac{\mu\,(1 + e)}{|C_3|} \qquad (ft) \qquad (2\text{-}122)$$

where e is the orbital eccentricity

V_A (Z(71)) Current apogee velocity (V_A). V_A is not defined for parabolic or hyperbolic orbits; under these conditions V_A is set equal to zero. For elliptic orbits:

$$V_A = \sqrt{\frac{\mu\,(1 - e)}{R_A}} \qquad (ft/sec) \qquad (2\text{-}123)$$

C_{3_M} (Z(72)) Orbital energy (C_{3_M}) expressed in metric units

$$C_{3_M} = C_3\,(CNV4)^2 \qquad (km^2/sec^2) \qquad (2\text{-}124)$$

where CNV4 is the conversion factor, 0.00030480061 kilometers per foot.

e (Z(73)) Orbital eccentricity (e)

$$e = \sqrt{1 + \frac{pC_3}{\mu}} \qquad (2\text{-}125)$$

where p is the orbital parameter in feet

p (Z(74)) Orbital parameter (p)

$$p = \frac{C^2}{\mu} \qquad \text{(ft)} \qquad (2\text{-}126)$$

where C is the Keplerian area constant in ft^2/sec

C (Z(75)) The Keplerian area constant (C). C is the angular momentum which is twice the time rate of area swept by the radius vector.

$$C = R\sqrt{V_{I_\phi}^2 + V_{I_\theta}^2} \qquad (ft^2/sec) \qquad (2\text{-}127)$$

η (Z(76)) The true anomaly (η). Let an argument of anomaly ARG_η be defined:

$$ARG_\eta = \text{ARCCOSINE}\left(\frac{[p/R - 1]}{e}\right) \qquad \text{(deg)} \qquad (2\text{-}128)$$

Then

$$\left.\begin{array}{lll} \text{if } \gamma_I \geq 0 & \eta = ARG_\eta \\ \text{if } \gamma_I < 0 & \eta = 360° - ARG_\eta \end{array}\right\} \text{(deg)} \qquad (2\text{-}129$$

τ (Z(77)) The orbital period (τ)

$$\tau = 2\pi a \sqrt{a/\mu} \qquad \text{(sec)} \qquad (2\text{-}130)$$

where a is the semi-major axis defined by

$$a = \frac{\mu}{|C_3|} \qquad \text{(ft)} \qquad (2\text{-}131)$$

ω_p (Z(91)) The argument of perigee (ω_p). ω_p is the angle between the ascending node and the perigee measured in the orbital plane, positive in the direction of orbital motion. For polar inclinations, ω_p is defined:

$$\left.\begin{aligned} \omega_p^* &= \theta - \eta \text{ when } V_I \text{ is directed northward} \\ \omega_p^* &= 180° - \theta - \eta \text{ when } V_I \text{ is directed southward} \end{aligned}\right\} \text{(deg)} \qquad (2\text{-}132)$$

For non-polar inclinations, the difference in longitude between the current position and the ascending node is given by:

$$\Delta\phi = \phi_N - \phi \qquad \text{(deg)} \qquad (2\text{-}133)$$

where $\Delta\phi$ is adjusted to be in the principal cycle, [0°, 360°). Let angular arguments C and C′ be defined by:

$$C'^{\dagger} = \text{ARCCOSINE } (\cos\theta \cos\Delta\phi) \qquad \text{(deg)} \qquad (2\text{-}134)$$

$$\left.\begin{aligned} C &= C' && \text{if } \theta < 0° \\ C &= 360° - C' && \text{if } \theta \geq 0° \end{aligned}\right\} \qquad \text{(deg)} \qquad (2\text{-}135)$$

Then ω_p is given by:

$$\omega_p^* = 360° - C - \eta \qquad \text{(deg)} \qquad (2\text{-}136)$$

R_p (Z(96)) Current geocentric perigee radius magnitude (R_p). R_p is expressed as:

$$R_p = \frac{p}{1+e} \qquad \text{(ft)} \qquad (2\text{-}137)$$

* The angular value is adjusted to be in the principal cycle, [0°, 360°), if "PRIF(K) = 0.," otherwise it is not adjusted.

† Here C′ is assumed to be between 0° and 180°, 0° ≤ C′ < 180°.

ϕ_p (Z(97)) The geocentric latitude of the perigee (ϕ_p)

$$\phi_p = \text{ARCCOS}(\sin i \sin \omega_p) \quad \text{(deg)} \quad (2\text{-}138)$$

ϕ_A (Z(98)) The geocentric latitude of the apogee (ϕ_A)

$$\phi_A = -\phi_p \quad \text{(deg)} \quad (2\text{-}139)$$

$R_{\oplus_A}$(REA) The surface radius of the earth at apogee and at perigee ($R_{\oplus_A}$). The surface is defined as an oblate spheroid (see Figure 2-14) with symmetry about the rotational axis and the equatorial plane. $R_{\oplus_A}$ is expressed by:

$$R_{\oplus_A} = \left[\frac{1-\sin^2\phi_p}{A^2} + \frac{\sin^2\phi_p}{B^2}\right]^{-1/2} \quad \text{(ft)} \quad (2\text{-}140)$$

h_p (Z(99)) The perigee altitude (h_p) as measured from the earth's oblate surface.

$$h_p = R_p - R_{\oplus_A} \quad \text{(ft)} \quad (2\text{-}141)$$

h_A (Z(100)) The apogee altitude (h_p) as measured from the earth's oblate surface.

$$h_p = R_A - R_{\oplus_A} \quad \text{(ft)} \quad (2\text{-}142)$$

The following parameters are used to target the orbiter to an orbit in which the apogee altitude, perigee altitude, and injection true anomaly has been specified. Let:

$$\left.\begin{array}{lll} SQ(3,1) & h_{A_{REQ}} = \text{required apogee altitude} & \text{(ft)} \\ SQ(3,2) & h_{P_{REQ}} = \text{required perigee altitude} & \text{(ft)} \\ SQ(3,3) & \eta_{REQ} = \text{required injection true anomaly} & \text{(deg)} \end{array}\right\} \quad (2\text{-}143)$$

The subscript REQ is used to denote the required value of any parameter in order to satisfy the input values of $h_{A_{REQ}}$, $h_{P_{REQ}}$, and η_{REQ}. These targeting parameters are defined by the following procedure.

RAT $\quad R_{A_{REQ}} = R_{\oplus_A} + h_{A_{REQ}}$ (ft) (2-144)

AT $\quad a_{REQ} = R_{\oplus_A} + \frac{1}{2}\left(h_{A_{REQ}} + h_{P_{REQ}}\right)$ (ft) (2-145)

ET $\quad e_{REQ} = \dfrac{R_{A_{REQ}}}{a_{REQ}} - 1$ (2-146)

SQ(3, 4) $\quad \cos \eta_{REQ}$

SQ(3, 5) $\quad \sin \eta_{REQ}$ (2-147)

RT $\quad R_{REQ} = \dfrac{R_{A_{REQ}}\left(1 - e_{REQ}\right)}{\left(1 + e_{REQ} \cos \eta_{REQ}\right)}$ (ft) (2-148)

HT $\quad h_{REQ} = R_{REQ} - R_{\oplus}$ (ft) (2-149)

VT $\quad V_{I_{REQ}} = \sqrt{\mu\left(\dfrac{2}{R_{REQ}} - \dfrac{1}{a_{REQ}}\right)}$ (ft/sec) (2-150)

VAT $\quad V_{A_{REQ}} = \sqrt{\mu\left(\dfrac{2}{R_{A_{REQ}}} - \dfrac{1}{a_{REQ}}\right)}$ (ft/sec) (2-151)

GAMT $\quad ARG_{\gamma} = \text{ARCCOSINE}\left(\dfrac{V_{A_{REQ}}\, R_{A_{REQ}}}{V_{I_{REQ}}\, R_{REQ}}\right)$ (2-152)

If $\sin(\eta_{REQ}) \geq 0 \quad \gamma_{I_{REQ}} = ARG_{\gamma}$

If $\sin(\eta_{REQ}) < 0 \quad \gamma_{I_{REQ}} = -ARG_{\gamma}$ (2-153)

Z(101) The difference between the current altitude and the required altitude (Δh) is:

$$\Delta h = h - h_{REQ} \tag{2-154}$$

Z(102) The difference between the current inertial flight path angle and the required inertial flight path angle ($\Delta\gamma_I$) is:

$$\Delta\gamma_I = \gamma_I - \gamma_{I_{REQ}} \tag{2-155}$$

Z(103) The difference between the current inertial velocity and the required inertial velocity (ΔV_I) is:

$$\Delta V_I = V_I - V_{I_{REQ}} \tag{2-156}$$

The following parameters define the instantaneous impact point. This point is the intersection of the conic orbit which is defined by the current trajectory conditions with the surface of the central body. These parameters although computed as part of the Orbital Elements are not computed automatically whenever the previously listed Orbital Elements are computed. These impact parameters are computed only if specifically called out by program input and only if

a. $R_p \leq 0.999B$

b. $R_A \leq B$

c. $0.0001 \leq e \leq 0.9999$

where R_p is the perigee radius of the instantaneous conic orbit.

Solution for oblate surface geometry (as opposed to assuming a spherical central body) is optional; however, it is noted that when the oblate solution is required, it is necessary to iterate on the value of the impact latitude (θ_{IMP}) to obtain the correct solution.

θ_{IMP} (Z(84)) Geocentric latitude of the instantaneous impact point (θ_{IMP}). There are two procedures for solution; the first procedure is used when the inclination (i) is nearly 90 degrees ($|i - 90°| < 0.000001$) and is presented in Table 2-4 while the second procedure is used for general non-90 degree inclination cases and is presented in Table 2-5 and Eq. (2-157)

Table 2-4. Latitude of the Instantaneous Impact Point for Polar Orbits

β_I	ARG_1	XKF1
$\beta_I = 0°$	$ARG_1 = \theta + \Delta\eta_{IMP}$	+1
$\beta_I = 180°$	$ARG_1 = \theta + \Delta\eta_{IMP}$	-1

ARG_1	θ_{IMP}
$\lvert ARG_1\rvert < 90°$	$\theta_{IMP} = ARG_1$
$90° \le \lvert ARG_1\rvert < 270°$	$\theta_{IMP} = (XKF1)(180°) - ARG_1$
$270° \le \lvert ARG_1\rvert$	$\theta_{IMP} = ARG_1 - (XKF1)(360°)$

where

ARG_1 is a temporary angular argument

XKF1 is a flag and/or coefficient.

θ_{IMP} is expressed

$$\theta_{IMP} = \mathrm{ARCSIN}\left[(XKF1)(XKF2)\sin i \sin D\right] \quad \text{(deg)} \qquad (2\text{-}157)$$

Table 2-5. Latitude of the Instantaneous Impact Point for Non-Polar Orbits

β_I	θ	XKF2	ϕ_A*	$\Delta\phi$*
$0° \le \beta_I \le 180°$	+	+1	$\phi_A = \phi_N - 180°$	$\Delta\phi = \phi_A - \phi$
$180° < \beta_I < 360°$	+	+1	$\phi_A = \phi_N - 180°$	$\Delta\phi = \phi - \phi_A$
$0° \le \beta_I \le 180°$	-	-1	$\phi_A = \phi_N$	$\Delta\phi = \phi_A - \phi$
$180° < \beta_I < 360°$	-	-1	$\phi_A = \phi_N$	$\Delta\phi = \phi - \phi_A$

C = ARCCOSINE ($\cos\theta \cos\Delta\phi$); $0° \le C < 180°$

$C - \Delta\eta$	D	XKF1	XKF3
$C - \Delta\eta_{IMP} \ge 0°$	$D = C - \Delta\eta_{IMP}$	+1	0
$0° > C - \Delta\eta_{IMP} > -180°$	$D = \Delta\eta_{IMP} - C$	-1	1
$-180° \ge C - \Delta\eta_{IMP}$	$D = \Delta\eta_{IMP} - C - 180°$	+1	2

where C and D are angular arguments

XKF1, XKF2, and XKF3 are flags and/or coefficients

$\Delta\phi$ is the difference in longitude between ϕ and ϕ_A

ϕ_A is the instantaneous longitude of the next nodal crossing.

*The angular value is adjusted to be in the principal cycle, [0°, 360°).

ϕ_{IMP} (Z(85)) Geocentric longitude of the instantaneous impact point (ϕ_{IMP})

$$\phi_{IMP} = \phi'_{IMP} - \omega \Delta t_{IMP} \quad \text{(deg)} \quad (2\text{-}158)$$

where

ϕ'_{IMP} is the longitude of the instantaneous impact point for a non-rotating central body ($\omega = 0$)

Δt_{IMP} is the time to impact from the current time.

$R_{\oplus_{IMP}}$ (Z(86)) Radius magnitude of the central body at the instantaneous impact point ($R_{\oplus_{IMP}}$).

$$R_{\oplus_{IMP}} = \left[\left(\frac{\cos \theta_{IMP}}{A}\right)^2 + \left(\frac{\sin \theta_{IMP}}{B}\right)^2\right]^{-1/2} \quad \text{(ft)} \qquad (2\text{-}159)$$

Δt_{IMP} (Z(87)) Time to impact from the current time (Δt_{IMP}). It is first necessary to evaluate the eccentric anomaly of the current position (E) and the impact point (E_{IMP})

$$\tan\left(\frac{E}{2}\right) = \sqrt{\frac{1-e}{1+e}} \tan\left(\frac{\eta}{2}\right) \Rightarrow 0 \le E < 2\pi \quad \text{(rad)} \qquad (2\text{-}160)$$

$$\tan\left(\frac{E_{IMP}}{2}\right) = \sqrt{\frac{1-e}{1+e}} \tan\left(\frac{\eta_{IMP}}{2}\right) \Rightarrow 0 \le E_{IMP} < 2\pi \quad \text{(rad)} \qquad (2\text{-}161)$$

Then

$$\Delta t_{IMP} = \left[\frac{E_{IMP} - E - e(\sin E_{IMP} - \sin E)}{2\pi}\right]\tau \quad \text{(sec)} \qquad (2\text{-}162)$$

ϕ'_{IMP} (Z(88)) Geocentric longitude of the instantaneous impact point for a non-rotating central body (ϕ'_{IMP}). As in the case of θ_{IMP}, there are two procedures for solution; the first procedure is used when the inclination (i) is nearly 90 degrees ($|i - 90°| < 0.000001$) and is presented in Table 2-6 while the second procedure is used for general non-90 degree inclination cases and is presented in Table 2-7.

Table 2-6. Longitude of the Instantaneous Impact Point for Polar Orbits and A Non-Rotating Central Body

ARG_1	ϕ'_{IMP} *
$\lvert ARG_1 \rvert < 90°$	$\phi'_{IMP} = \phi$
$90° \leq \lvert ARG_1 \rvert < 270°$	$\phi'_{IMP} = \phi + 180°$
$270° \leq \lvert ARG_1 \rvert$	$\phi'_{IMP} = \phi$

where ARG_1 is defined in Table 2-4 and Eq. (2-163)
* The angular value is adjusted to be in the principal cycle, [0°, 360°).

Table 2-7. Longitude of the Instantaneous Impact Point for Non-Polar Orbits and A Non-Rotating Central Body

$$\text{Let } ARG_\phi = \text{ARCCOSINE}\left(\frac{\cos D}{\cos \theta_{IMP}}\right) \quad ; \; 0° \leq ARG_1 < 180°$$

XKF3	β_I	$\Delta\phi_{IMP}$
0	$0° \leq \beta_I \leq 180°$	$\Delta\phi_{IMP} = -ARG_\phi$
0	$180° < \beta_I$	$\Delta\phi_{IMP} = ARG_\phi$
1	$0° \leq \beta_I \leq 180°$	$\Delta\phi_{IMP} = A$[illegible]$_\phi$
1	$180° < \beta_I$	$\Delta\phi_{IMP} = -ARG_\phi$
2	$0° \leq \beta_I \leq 180°$	$\Delta\phi_{IMP} = ARG_\phi + 180°$
2	$180° < \beta_I$	$\Delta\phi_{IMP} = -ARG_\phi - 180°$

where D and XKF3 are defined in Table 2-5

$\Delta\phi_{IMP}$ is the difference in longitude between ϕ'_{IMP} and ϕ_A

Then ϕ'_{IMP} is expressed*

$$\phi'_{IMP} = \phi_A + \Delta\phi_{IMP} \qquad \text{(deg)} \qquad (2\text{-}163)$$

$\Delta\eta_{IMP}$ (Z(89)) Central angle from the current position to impact ($\Delta\eta_{IMP}$)

$$\Delta\eta_{IMP} = \eta_{IMP} - \eta \qquad \text{(deg)} \qquad (2\text{-}164)$$

where η_{IMP} is the true anomaly of the impact point

η_{IMP} (Z(90)) True anomaly of the impact point (η_{IMP})

$$ARG_\eta = \text{ARCCOSINE}\left(\frac{\left[\frac{p}{R_{\oplus_{IMP}}} - 1\right]}{e}\right) \qquad \text{(deg)} \qquad (2\text{-}165)$$

$$0° \leq ARG_\eta \leq 180°$$

Then

$$\eta_{IMP} = 360° - ARG_\eta \qquad \text{(deg)} \qquad (2\text{-}166)$$

Position, Velocity, and Acceleration Vector Components. There are two options (A and B) currently available in GTSM which are used to determine vector components of the position, velocity, and acceleration vectors in the Earth-Fixed Coordinate System or in an Inertial Coordinate System. Figure 2-17 defines the Inertial Coordinate System graphically.

The $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ Inertial Coordinate System is defined at the initiation of the trajectory simulation and is a right-handed orthogonal system with the third axis ($\hat{J}_3$ axis) passing through the meridian which is defined by ϕ_L at a specific angle ϵ from the I_3 axis of the Primary Inertial Coordinate System. The direction of the first axis ($\hat{J}_1$ axis) is defined by a relative azimuth angle β_ϵ which is also specified. The second axis ($\hat{J}_2$ axis) completes the right-handed orthogonal system.

*The angular value is adjusted to be in the principal cycle [0°, 360°).

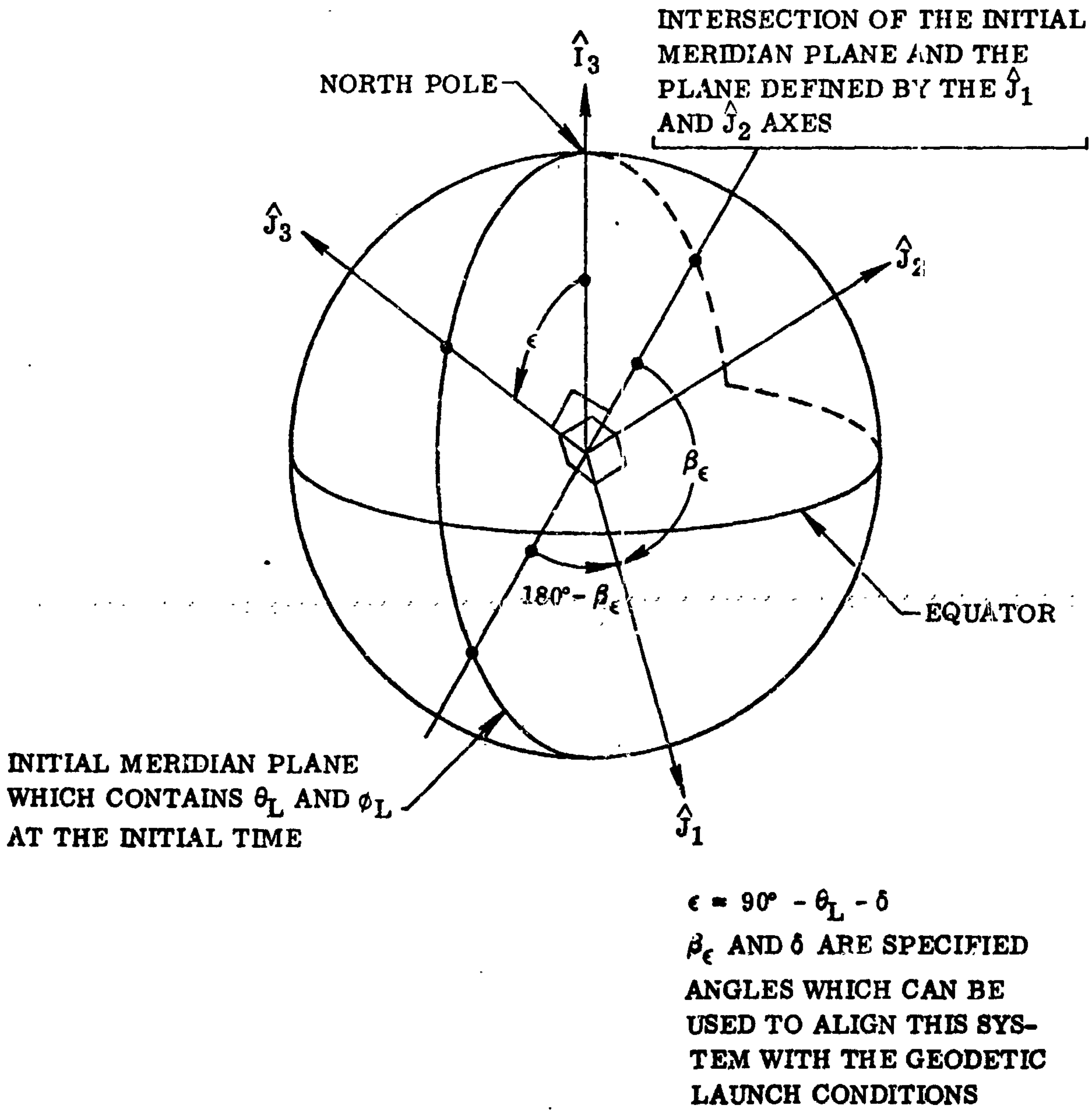


Figure 2-17. The $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ Inertial Coordinate System

The vectors which are to be resolved into components are normally expressed in either the Relative Velocity or the Inertial Velocity* Coordinate System; consequently, it is necessary to transform these vectors into the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ system. Four transformations are required; these are:

a. The Relative Velocity System to the Local System

b. The Inertial Velocity System to the Local System

c. The Local System to the Earth-Fixed System

d. The Earth-Fixed System to the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ Inertial System

The first transformation which transforms a vector from the $\hat{i}_\beta - \hat{i}_V - \hat{i}_\gamma$ system to $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ system is defined by the transformation matrix [M-5] of Eq. (M-5). The second transformation which transforms a vector from the $\hat{i}_{\beta_I} - \hat{i}_{V_I} - \hat{i}_{\gamma_I}$ system to the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ system is defined by the transformation matrix [M-6] below. [M-6] is obtained directly from [M-5] by replacing β, V, and γ with β_I, V_I, and γ_I. For a general vector $\bar{A}$ which is expressed in the Inertial Velocity System, it is transformed to the Local System by [M-6].

$$\begin{pmatrix} A_\phi \\ A_\theta \\ A_R \end{pmatrix} = \begin{pmatrix} \cos\beta_I & \sin\beta_I \cos\gamma_I & -\sin\beta_I \sin\gamma_I \\ -\sin\beta_I & \cos\beta_I \cos\gamma_I & -\cos\beta_I \sin\gamma_I \\ 0 & \sin\gamma_I & \cos\gamma_I \end{pmatrix} \begin{pmatrix} A_{\beta_I} \\ A_{V_I} \\ A_{\gamma_I} \end{pmatrix} \qquad \text{(M-6)}$$

Noting that, for orthogonal coordinate system transformations, the inverse matrix is equal to the transpose matrix, Eq. (M-3) (and the inverse of matrix [M-3]) yields the third transformation matrix [M-7], which transforms a vector from the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ system to the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ system. For the general vector $\bar{A}$, this transformation is:

$$\begin{pmatrix} A_{e_1} \\ A_{e_2} \\ A_{e_3} \end{pmatrix} = \begin{pmatrix} -\sin\phi & -\cos\phi \sin\theta & \cos\phi \cos\theta \\ \cos\phi & -\sin\phi \sin\theta & \sin\phi \cos\theta \\ 0 & \cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} A_\phi \\ A_\theta \\ A_R \end{pmatrix} \qquad \text{(M-7)}$$

The fourth transformation which transforms a vector from the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ system to the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ system is derived by assuming that the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ axes are initially

*Similar to the Relative Velocity System except the axes $(\hat{i}_{\beta_I} - \hat{i}_{V_I} - \hat{i}_{\gamma_I})$ are defined by $\bar{R}$ and $\bar{V}_I$ rather than $\bar{R}$ and $\bar{V}$.

aligned in the same direction as the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ axes, respectively, and then performing three sequenced rotations which result in the required orientation of $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$. Referring to Figure 2-18, let $\hat{J}_1'''$, $\hat{J}_2'''$, $\hat{J}_3'''$ denote the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ axes in the initially assumed alignment with the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ axes. The three rotations are defined:

a. The first rotation is a rotation of the $\hat{J}_1''' - \hat{J}_2''' - \hat{J}_3'''$ about the $\hat{J}_3'''$ axis through an angle C to a new orientation $\hat{J}_1'' - \hat{J}_2'' - \hat{J}_3''$. C is the angle which results in placing $\hat{J}_1''$ at a longitude of ϕ_L and is given by:

$$C = \phi_L - \omega t \qquad \text{(deg)} \qquad (2\text{-}167)$$

where t is the absolute time

b. The second rotation is about the $\hat{J}_2''$ axes through an angle ϵ to a new orientation $\hat{J}_1' - \hat{J}_2' - \hat{J}_3'$ where ϵ is:

$$\epsilon = 90° - \theta_L - \delta \qquad \text{(deg)} \qquad (2\text{-}168)$$

where δ_L is defined by program input (see "GTIP" in Section 4.2.3.1

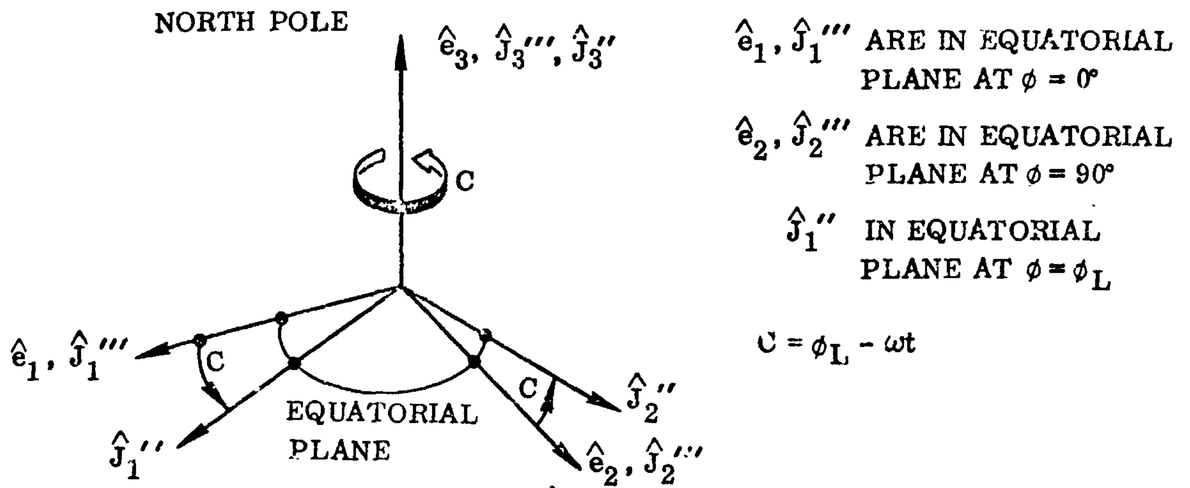
c. The final rotation is about the $\hat{J}_3'$ axis through an angle D to the required orientation $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$. D is given by

$$D = 180° - \beta_\epsilon \qquad \text{(deg)} \qquad (2\text{-}169)$$

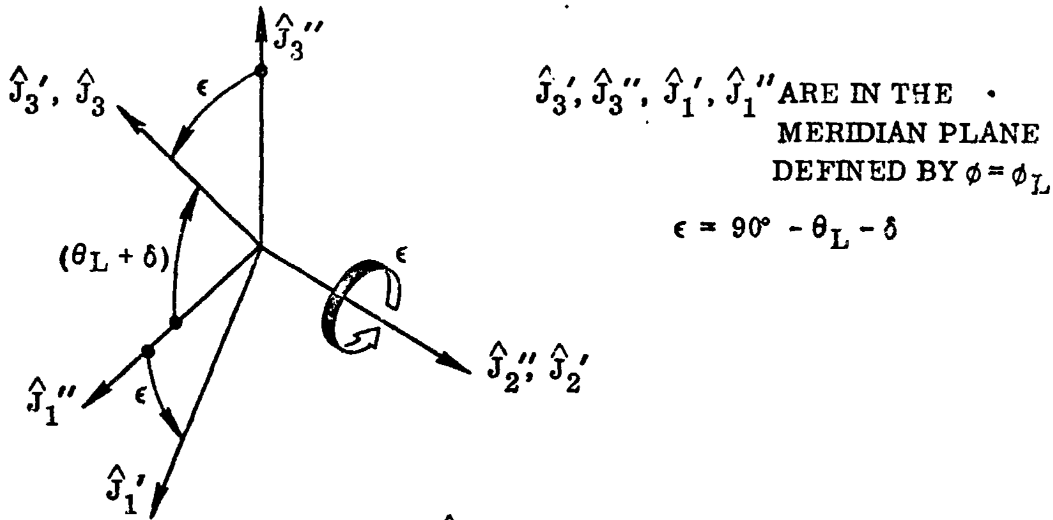
where β_ϵ is defined by program input (see "AZMI" in Section 4.2.3.1

If $\bar{A}$ is any vector which is expressed in the $\hat{e}_1 - \hat{e}_2 - \hat{e}_3$ system, then successive applications of the elemental coordinate transformation matrices (A-M3), (A-M2), and (A-M3) in that order for rotation angles of C, ϵ, and D, respectively, yields $\bar{A}$ expressed in the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ system and defines the transformation matrix [M-8]:

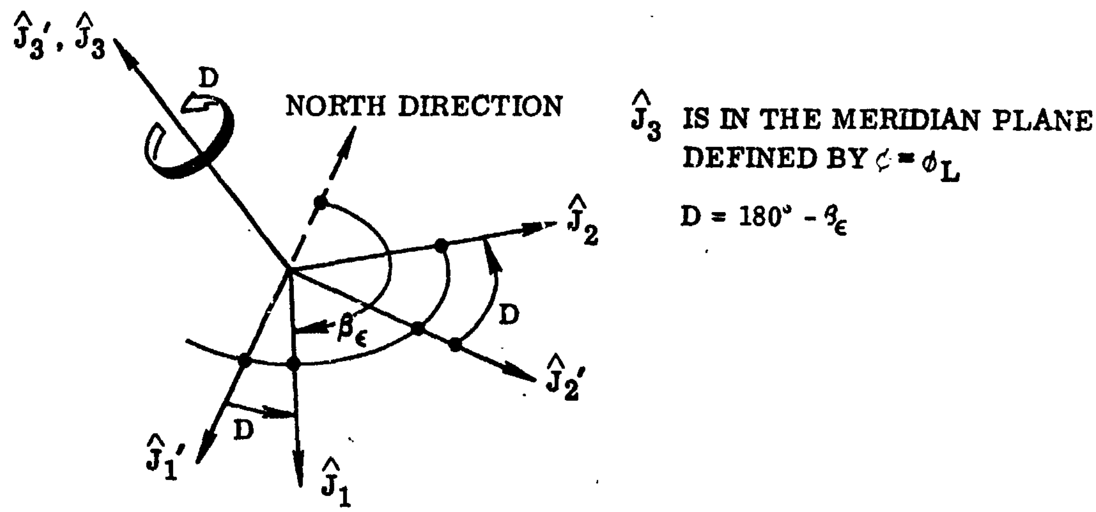
$$\begin{pmatrix} A_{J_1} \\ A_{J_2} \\ A_{J_3} \end{pmatrix} = \begin{pmatrix} \cos D & \sin D & 0 \\ -\sin D & \cos D & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \epsilon & 0 & -\sin \epsilon \\ 0 & 1 & 0 \\ \sin \epsilon & 0 & \cos \epsilon \end{pmatrix} \begin{pmatrix} \cos C & \sin C & 0 \\ -\sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} A_{e_1} \\ A_{e_2} \\ A_{e_3} \end{pmatrix}$$



FIRST ROTATION: ROTATION ABOUT $\hat{J}_3'''$ THROUGH C



SECOND ROTATION: ROTATION ABOUT $\hat{J}_2''$ THROUGH ϵ



THIRD ROTATION: ROTATION ABOUT $\hat{J}_3'$ THROUGH D

Figure 2-18. Transformation from Earth Fixed Axis to $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ Inertial Axes

which upon carrying out the multiplication becomes:

$$\begin{pmatrix} A_{J_1} \\ A_{J_2} \\ A_{J_3} \end{pmatrix} = \begin{pmatrix} [\cos C \cos \epsilon \cos D - \sin C \sin D] & [\sin C \cos \epsilon \cos D + \cos C \sin D] & [-\sin \epsilon \cos D] \\ [-\cos C \cos \epsilon \sin D - \sin C \cos D] & [-\sin C \cos \epsilon \sin D + \cos C \cos D] & [\sin \epsilon \sin D] \\ [\cos C \sin \epsilon] & [\sin C \sin \epsilon] & [\cos \epsilon] \end{pmatrix} \begin{pmatrix} A_{e_1} \\ A_{e_2} \\ A_{e_3} \end{pmatrix} \quad \text{[M-8]}$$

The angle δ is defined by program input. There is an option in this definition to internally set δ equal to the geodetic tip angle (δ_T) at the geocentric latitude θ_L. Use of this option results in orienting the $\hat{J}_3$ axis parallel to this geodetic vertical. The expression for δ_T in terms of θ_L, the equatorial radius (A) of the central body, and the polar radius (B) of the central body is readily derived from the geometry which is illustrated in Figure 2-19. This cross section is described by assuming the properties of an ellipse, consequently:

$$\frac{x^2}{A^2} + \frac{y^2}{B^2} = 1 \tag{2-170}$$

The slope of the tangent T at point S is $\frac{dx}{dy}$ and is obtained directly from Eq. (2-170) viz.,

$$\text{slope of tangent at S} = \frac{dx}{dy} = -\frac{A^2 y}{B^2 x} \tag{2-171}$$

The slope of the line $\overline{PS}$ (perpendicular to the tangent), at S is $-1\Big/\left(\frac{dx}{dy}\right)$, consequently:

$$\text{slope of } \overline{PS} = \frac{-1}{\left(\frac{dx}{dy}\right)} = \tan \theta_g \tag{2-172}$$

The line $\overline{OS}$ has a slope:

$$\text{slope of } \overline{OS} = \frac{y}{x} = \tan \theta_L \tag{2-173}$$

By combining Eq. (2-171), (2-172), and (2-173), θ_g is expressed in terms of θ_L:

$$\tan \theta_g = \frac{A^2}{B^2} \tan \theta_L \tag{2-174}$$

δ_T can now be obtained from Eq. (2-174) and the geometry of Figure 2-19 as:

$$\delta_T = \theta_g - \theta_L$$

$$\therefore \delta_T = \text{ARCTAN}\left[\frac{A^2}{B^2} \tan \theta_L\right] - \theta_L \tag{2-175}$$

Since Eq. (2-175) is defined by tangent functions, and since δ_T is a first or fourth quadrant angle, the correct algebraic sign is automatically determined when evaluating it.

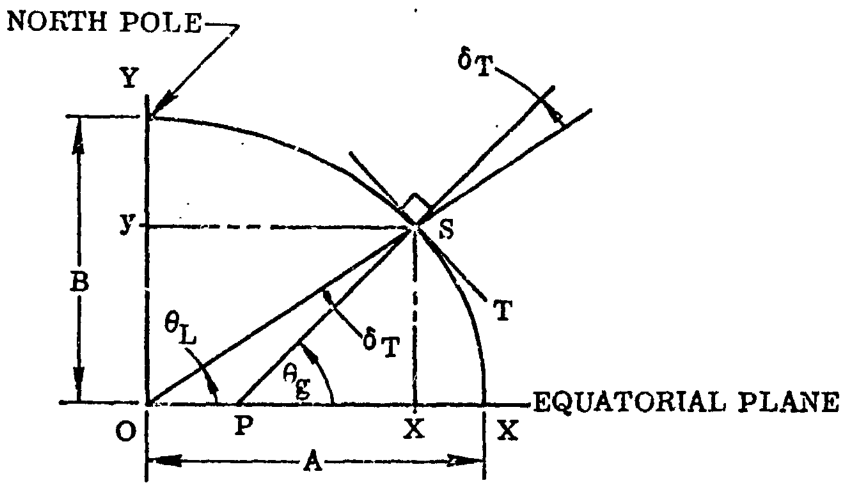


Figure 2-19. Geodetic Tip Angle

<u>Option A</u>. Option A determines the vector components of the geocentric radius vector $\bar{R}$ and the relative velocity vector $\bar{V}$ in the Earth-Fixed Coordinate System (see Figure 2-6), and the net acceleration vector, $\bar{a}$ in the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ Inertial System. The net acceleration vector includes all accelerations acting on the vehicle — aerodynamic, propulsive, and gravitational.

Let R_1, R_2, and R_3 denote the components of $\bar{R}$ along the $\hat{e}_1$, $\hat{e}_2$, and $\hat{e}_3$ axes, respectively. The associated program internal designations are Z(53), Z(54), and Z(55), respectively. Expression of $\bar{R}$ as

$$\bar{R} = (0)\hat{i}_\phi + (0)\hat{i}_\theta + (R)\hat{i}_R$$

and application of transformation matrix [M-7] yields:

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} = [M-7] \begin{pmatrix} 0 \\ 0 \\ R \end{pmatrix} \qquad \text{(ft)} \qquad (2\text{-}176)$$

Let V_1, V_2, and V_3 denote the components of $\bar{V}$ along the $\hat{e}_1$, $\hat{e}_2$, and $\hat{e}_3$ axes, respectively, with corresponding respective associated program internal designations Z(56), Z(57), and Z(58). Then for

$$\bar{V} = (0)\,\hat{i}_\beta + (V)\,\hat{i}_V + (0)\,\hat{i}_\gamma$$

successive application of transformation matrices [M-5] and [M-7] yields:

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = [M-7]\,[M-5] \begin{pmatrix} 0 \\ V \\ 0 \end{pmatrix} \qquad \text{(ft/sec)} \qquad (2\text{-}177)$$

The total applied force (does not include the gravitational force) vector is expressed in the Relative Velocity System as:

$$\bar{F}_{APPLIED} = F_\beta\,\hat{i}_\beta + F_V\,\hat{i}_V + F_\gamma\,\hat{i}_\gamma \qquad \text{(lb)} \qquad (2\text{-}178)$$

Multiplying Eq. (2-178) by $\left(\frac{g_0}{W}\right)$ yields the applied force acceleration vector ($\bar{f}$) with components f_β, f_V, and f_γ where g_0 is the reference gravitational acceleration at which weight is measured. $\bar{f}$ is then transformed into the $\hat{i}_\phi$ - $\hat{i}_\theta$ - $\hat{i}_R$ system by means of the transformation matrix [M-5], thus, yielding components f_ϕ, f_θ, and f_R:

$$\begin{pmatrix} f_\phi \\ f_\theta \\ f_R \end{pmatrix} = [M-5] \begin{pmatrix} f_\beta \\ f_V \\ f_R \end{pmatrix} \qquad \text{(ft/sec)} \qquad (2\text{-}179)$$

The total net acceleration vector ($\bar{a}$) is obtained by adding the components of the gravitational acceleration vector $\bar{g}$ (with components g_θ and $-g_R$) to the total applied acceleration vector. This is accomplished easily in the $\hat{i}_\phi - \hat{i}_\theta - \hat{i}_R$ frame. The resulting components are then transformed to the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ Inertial System by successive application of transformation matrices [M-7] and [M-8], thus, yielding the net acceleration vector expressed in the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ system. Let a_1, a_2, and a_3 denote the components of $\bar{a}$ with corresponding respective program internal designations Z(59), Z(60), and Z(61); then:

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = [M\text{-}8][M\text{-}7] \begin{pmatrix} f_\phi \\ f_\theta + g_\theta \\ f_R - g_R \end{pmatrix} \qquad (\text{ft/sec}^2) \qquad (2\text{-}180)$$

Option B. Option B defines the vector components of the geocentric radius vector $\bar{R}$, the inertial velocity vector V_I, the total applied force acceleration vector $\bar{f}$, and the gravitational vector g in the $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ Inertial Coordinate System. The total applied force acceleration vector with components f_β, f_V, f_γ is obtained by multiplying the total applied force vector $\bar{F}_{APPLIED}$ (Eq. (2-180)) by $\left(\frac{g_0}{W}\right)$. This vector does not include the acceleration due to gravity. The vector components are designated according to Table 2-8.

Table 2-8. Vector Component Designations

Vector	Components In The Frame In Which Initially Expressed	Components In The $\hat{J}_1 - \hat{J}_2 - \hat{J}_3$ Inertial Frame	Corresponding Program Inertial Designations
$\bar{R}$	R ($\hat{i}_R$ axis)	R_1, R_2, R_3	Z(53), Z(54), Z(55)
$\bar{V}_I$	V_I ($\hat{i}_{V_I}$ axis)	V_{I1}, V_{I2}, V_{I3}	Z(56), Z(57), Z(58)
$\bar{f}$	f_β, f_V, f_γ	f_1, f_2, f_3	Z(59), Z(60), Z(61)
$\bar{g}$	g_θ, $-g_R$	g_1, g_2, g_3	Z(62), Z(63), Z(64)

Then by successive applications of appropriate transformation matrices of [M-5], [M-6], [M-7], and [M-8] the vector components are obtained; succinctly:

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} = [M\text{-}8][M\text{-}7] \begin{pmatrix} 0 \\ 0 \\ R \end{pmatrix} \qquad \text{(ft)} \qquad (2\text{-}181)$$

$$\begin{pmatrix} V_{I_1} \\ V_{I_2} \\ V_{I_3} \end{pmatrix} = [M\text{-}8][M\text{-}7][M\text{-}6] \begin{pmatrix} 0 \\ V_I \\ 0 \end{pmatrix} \qquad \text{(ft/sec)} \qquad (2\text{-}182)$$

$$\begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = [M\text{-}8][M\text{-}7][M\text{-}5] \begin{pmatrix} f_\beta \\ f_V \\ f_\gamma \end{pmatrix} \qquad \text{(ft/sec}^2\text{)} \qquad (2\text{-}183)$$

$$\begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} = [M\text{-}8][M\text{-}7] \begin{pmatrix} 0 \\ g_\theta \\ -g_R \end{pmatrix} \qquad \text{(ft/sec}^2\text{)}$$

<u>Position in Radar Coordinates</u>. It is possible to compute the current position in radar coordinates from the initial trajectory coordinates (θ_L, ϕ_L, R_L) and, if required, the currently defined target coordinates (θ_T, ϕ_T, R_T). R_L is the radius magnitude at the initial position.

R_T (WQ(10, M)) Radius magnitude of the current target radar site

$$R_T = R_{\oplus T} + h_T \qquad \text{(ft)} \qquad (2\text{-}184)$$

where: $R_{\oplus T}$ is the surface radius magnitude at θ_T

h_T is the altitude of the radar site

AZ_L (Z(52)) Azimuth of the vehicle measured at the initial trajectory coordinates (AZ_L).

$$AZ_L \equiv \beta_R \qquad \text{(deg)} \qquad (2\text{-}185)$$

See Eq. (2-106) and (2-107) in Section 2.2.2.9.

SR_L (Z(79)) Slant range of the vehicle measured at the initial trajectory coordinates (SR_L).

$$\overline{SR}_L = R\,\hat{R} - R_L\,\hat{L} \qquad \text{(ft)} \qquad (2\text{-}186)$$

$$SR_L = |\overline{SR}_L| \qquad \text{(ft)} \qquad (2\text{-}187)$$

EA_L (Z(80)) Elevation angle of the vehicle measured at the initial trajectory coordinates (EA_L). Let

$$ARG_X = \text{ARCCOSINE}\left(\frac{\hat{L} \cdot \overline{SR}_L}{SR_L}\right) \qquad \text{(deg)} \qquad (2\text{-}188)$$

Then

$$EA_L = 90^\circ - ARG_X \qquad \text{(deg)} \qquad (2\text{-}189)$$

AZ_T (Z(81)) Azimuth of the vehicle measured at the current target radar site coordinates (AZ_T). AZ_T is determined in a manner similar to the procedure used to evaluate β_R except that $\hat{T}$ replaces $\hat{L}$ in Eq. (2-106) and (2-107)

$$ARG_{AZ} = \text{ARCCOSINE}\left(\frac{(\hat{T} \times \hat{R}) \cdot \hat{W}}{|\hat{T} \times \hat{R}|}\right) \qquad \text{(deg)} \qquad (2\text{-}190)$$

Then from geometry, AZ_T is given by

$$\left.\begin{array}{l} \text{if } [\hat{T} \times \hat{R}] \cdot \hat{e}_3 \geq 0, \; AZ_T = ARG_{AZ} \\ \text{if } [\hat{T} \times \hat{R}] \cdot \hat{e}_3 < 0, \; AZ_T = 360^\circ - ARG_{AZ} \end{array}\right\} \text{(deg)} \qquad (2\text{-}191)$$

SR_T (Z(82)) Slant range of the vehicle measured at the current target radar site coordinates (SR_T)

$$\overline{SR}_T = R\hat{R} - R_T\hat{T} \quad \text{(ft)} \qquad (2\text{-}192)$$

$$SR_T = |\overline{SR}_T| \quad \text{(ft)} \qquad (2\text{-}193)$$

EA_T (Z(83)) Elevation angle of the vehicle measured at the current target radar site coordinates (EA_T). Let

$$ARG_X = \text{ARCCOSINE}\left(\frac{\hat{T} \cdot \overline{SR}_T}{SR_T}\right) \quad \text{(deg)} \qquad (2\text{-}194)$$

Then

$$EA_T = 90° - ARG_X \quad \text{(deg)} \qquad (2\text{-}195)$$

AZ_V (Z(93)) Azimuth of the target site as seen from the vehicle (AZ_V). AZ_V is determined simularly to the procedure used to evaluate β_R (Eq. (2-190) and (2-191))except that the $\hat{T}$ and $\hat{R}$ are interchanged.

$$ARG_{AZV} = \text{ARCCOSINE}\left(\frac{[(\hat{R} \times \hat{T}) \cdot \hat{W}]}{|\hat{R} \times \hat{T}|}\right) \quad \text{(deg)} \qquad (2\text{-}196)$$

From the geometry, AZ_V is given by:

$$\left.\begin{array}{ll} \text{If } [\hat{R} \times \hat{T}] \cdot \hat{e}_3 \geq 0,\ AZ_V = ARG_{AZV} & \text{(deg)} \\ \text{If } [\hat{R} \times \hat{T}] \cdot \hat{e}_3 < 0,\ AZ_V = 360° - AZ_V & \text{(deg)} \end{array}\right\} \qquad (2\text{-}197)$$

ΔAZ_V (Z(94)) The difference between the azimuth of the target site from the vehicle and the azimuth of the vehicle (ΔAZ_V) (2-198)

$$\Delta AZ_V = AZ_V - \beta \quad \text{(deg)}$$

There is an option to increment ΔAZ_V by $\pm n$ 360° for $0 \leq n \leq 10$ in order to restrict the value of $|\Delta AZ_V|$ to be less than 360 deg.

In any event, tests are made to verify that ΔAZ_V has the correct algebraic sign (e.g., if $\Delta AZ_V = 0$ is being used to terminate a simulation section, the value of ΔAZ_V should approach zero) and is within 300 deg of zero. These tests are:

If $\dot{\beta} < 0$ and $\Delta AZ_V > 60$ deg, then ΔAZ_V is decremented by 360 deg.

$$\Delta AZ_V = \Delta AZ_V - 360 \qquad \text{(deg)} \qquad (2\text{-}199)$$

If $\dot{\beta} > 0$ and $\Delta AZ_V < -60$ deg, then ΔAZ_V is incremented by 360 deg.

$$\Delta AZ_V = \Delta AZ_V + 360 \qquad \text{(deg)} \qquad (2\text{-}200)$$

RNG_T (Z(95)) The surface range in nautical miles between the vehicle and the target site or landing site (TNG_T)

$$RNG_T = \left(\frac{T_{MISS}}{RAD}\right)\left(\frac{R_T - h_T}{CNV3}\right) \qquad \text{(n.mi.)} \qquad (2\text{-}201)$$

where:

CNV3 is the conversion factor which defines the number of feet per nautical mile (6076.1033 ft/n.mi.).

RAD is the radian to degree conversion factor; 57.2957795 degrees/radian

2.3 SYNTHESIS TECHNIQUES

The overall organization of the SSSP is shown in Figure 2-20, and also illustrates the major iteration process that exists between the WTVOL and GTSM subprograms. The additional iterations that exist due to option capabilities provided by the SSSP are contained within this major iteration process. The answer sought for a given run (or major iteration) is a vehicle and trajectory combination which will deliver a specified payload to a specified parking orbit. (This specified payload is provided by internal iterations to the WTVOL subprogram if the specified gross liftoff weight (GLOW) option is utilized.) For each run the following basic items are fixed: payload weight or GLOW, number of main engines per stage, injection conditions (V_f, h_f, and γ_f), staging γ or staging dynamic pressure, specific impulses, and ΔV allowances for post-injection maneuvering. Commonality options also fix several booster characteristics (weights, thrust levels, etc.) in terms of calculated orbiter values. Within the WTVOL subprogram it is thus convenient to design (size) the orbiter first, then the booster. For the orbiter the weight and volume computations are based on fixed thrust/weight or thrust, payload and mass ratio μ_o. The answers are gross weight and volume. The booster computations are based on fixed thrust/weight or thrust, payload and mass ratio μ_B, its payload being the computed orbiter gross weight. In order to size the booster and orbiter, it is therefore necessary to know μ_B and μ_o. They are treated as two independent parameters by the iteration logic in the SSSP, and are determined in order to satisfy the vehicle (size constraints) and trajectory (characteristic velocity) conditions imposed by the groundrules. Initial estimates of μ_B and total characteristic velocity (to perigee insertion) are input (the initial estimate of μ_o is computed internally) and adjustments are made after each WTVOL/GTSM computation until a vehicle/trajectory solution is obtained to within input-specified tolerances.

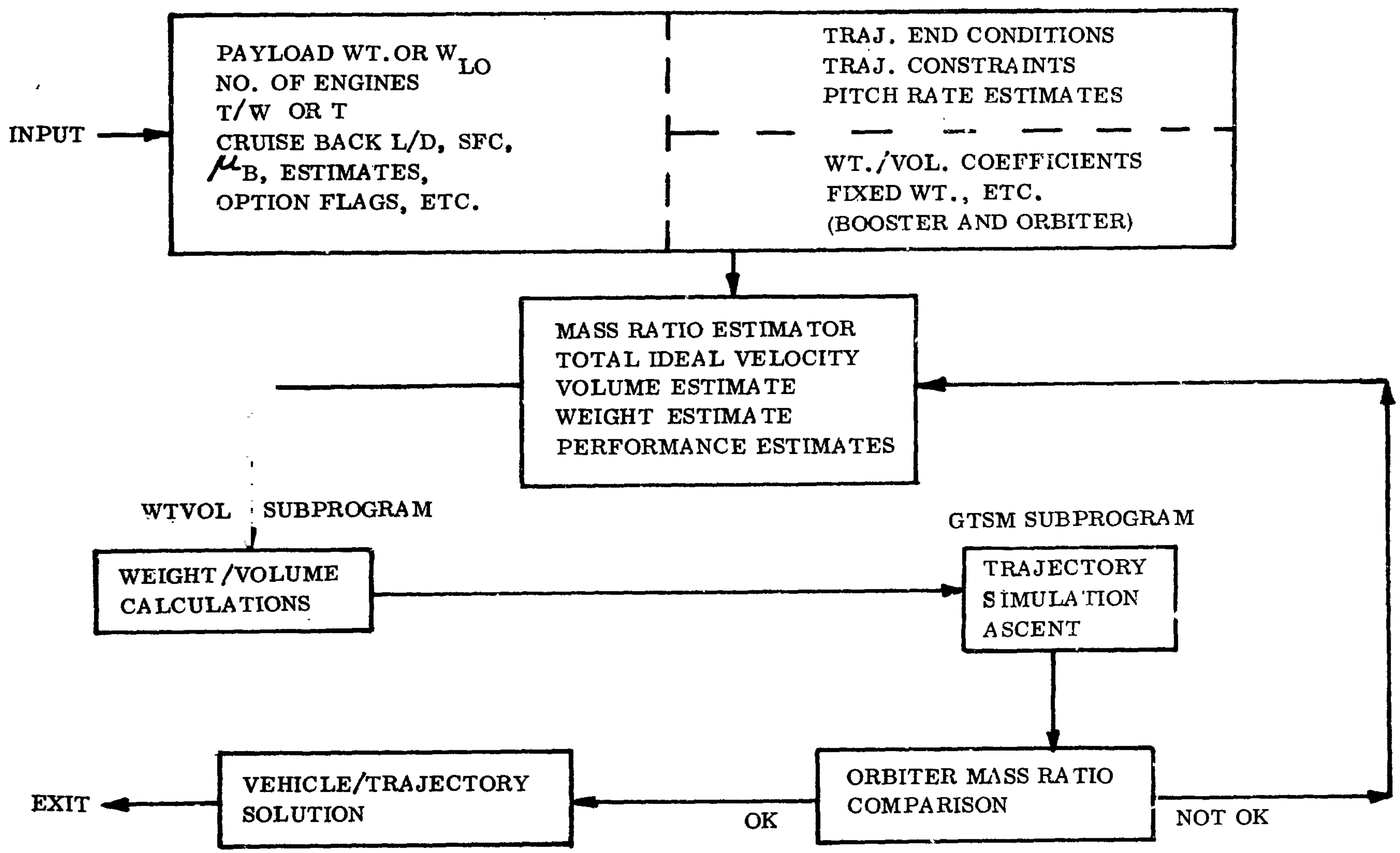


Figure 2-20. SSSP Organization

The primary function of the GTSM subprogram in the synthesis procedure is to provide a characteristic velocity requirement update from the initial estimate (by determining the velocity losses on the trajectory which change as the vehicle design varies), thereby determining the mass ratios. (A secondary function of GTSM is to provide booster subsonic cruise data for adjustment of the booster airbreathing engine fuel requirements).

The organization of the SSSP was substantially influenced by the large storage requirements of the component routines (WTVOL and GTSM), which necessitated usage of the OVERLAY feature of the Univac 1108 computer. This is discussed in Section 3 of this report, Programming Discussion. It is introduced here because of its impact on the computational sequence. Basically, the OVERLAY feature allows large parts of the program to be temporarily stored on peripheral memory equipment, unavailable for computations, while other parts (called OVERLAYS) are used in the central processor. The WTVOL and GTSM routines cannot be utilized simultaneously; neither can the SSSP input and program control routines be used with either of the other two. Transferral of these OVERLAYS between central and peripheral equipment is fairly expensive in computer time. Consequently, it is economical to minimize the number of transfers from one OVERLAY to the other. Maximum useful information must be obtained from each OVERLAY before transferring to another. This influences the type of iteration logic used to satisfy the vehicle size constraints and velocity requirements. The procedure chosen, which has proven to be very efficient, consists of nested iteration loops: an <u>outer loop</u>, requiring one transfer of each OVERLAY, whose purpose is to satisfy the velocity requirement, and a series of <u>inner loops</u>, handled internal to the WTVOL subprogram, whose purpose is to satisfy certain vehicle constraints imposed on the problem such as requiring a specified gross liftoff weight for the system. The operation of these loops are discussed in the following subsections in terms of the operational capabilities and synthesis interface techniques of the SSSP. The WTSCH subroutine, which contains the data bank of weight scaling laws (described in Volume II of this report), and the trajectory subprogram (GTSM) will be treated as two "black-boxes" in this section. The synthesis interface parameters which drive these two basic portions of the program will be discussed in terms of synthesis driver input parameters, the calculated items output from these two basic program blocks, and any basic input parameters which are required from the weight/volume data blocks in order to describe the synthesis iteration process or option capability of the SSSP. These input data blocks are discussed in Sections 3 and 4 of this report however, in the following sections when a parameter is an input to the synthesis driver data block, $DATA2, the input parameter will be denoted in capital letters identical to the actual input acronym.

2.3.1 BASIC SYNTHESIS ITERATION

The important fixed parameters for a SSSP case are payload, ascent trajectory terminal conditions, stage specific impulses, main-engine thrust levels (or ignition thrust-to-weight ratios, T/W's), and staging flight path angle or staging dynamic pressure. The most important input estimate is the required total characteristic velocity, ΔV (input as IDVEL*), from liftoff to parking orbit injection. This quantity is characteristic of a given mission, and is usually known fairly well after a few runs with a particular configuration. It varies primarily with liftoff thrust-to-weight. The gross vehicle size is strongly dependent on ΔV. A good estimate of IDVEL results in fewer program iterations, since the initial vehicle is sized for approximately the correct velocity requirement; the trajectory computations merely refine the velocity requirement by accounting for the velocity losses, which vary rather weakly with vehicle stage sizes and characteristics.

*In succeeding sections synthesis input parameters will be capitalized and in the form input to the $DATA2 input data block.

The computational flow (as illustrated in Figure 2-20) is summarized as follows:

1. Input data to basic data blocks: fixed input quantities and estimates for IDVEL, main impulse mass ratios for the booster and orbiter (μ_B, μ_o) and gross stage weight for the booster and orbiter (W_{G_B}, W_{G_O}).

2. Process input data and adjust μ_o estimate to agree with IDVEL estimate.
3. In WTVOL, size the vehicle:
   a. compute W_{G_O}, using fixed T/W_o or thrust, μ_o, and payload
   b. compute W_{G_B}, using fixed T/W_B or thrust, μ_B, and booster payload $= W_{G_O}$
   c. compare vehicle with sizing constraints imposed by input options if utilized and, if not within specified tolerances, adjust control parameters and repeat step (a) cycle
   d. prepare GTSM data from WTVOL output
4. In GTSM, simulate ascent trajectory with fixed μ_B, determine μ_o* required to reach the specified terminal velocity (V_f) at orbit insertion.
5. In synthesis driver, test the error $\mu_o - \mu_o$* with specified tolerance,

   a. if acceptable, set termination flag and cycle through WTVOL and GTSM for printout
   b. if not acceptable, adjust μ_o and repeat Step 3 cycle.

The adjustment made to the input value of μ_o in Step 2 above is made as follows:

$$\Delta V_o = \text{IDVEL} - I_B \, g_o \, \ell n \, \mu_B$$

and

$$\mu_o = e^{\Delta V_o / (I_{V_o} \, g_o)}$$

where ΔV_o = characteristic velocity increment estimated for the orbiter

g_o = 32.2

I_B = effective specific impulse of the booster

= (1 - PERISP)*ISLB(1) + PERISP*IVACB(2)

PERISP = input scaling factor (≈.80) which accounts for the impulse lost due to atmospheric pressure on the nozzle exit areas.

ISLB(1) = input value of the sea level specific impulse for booster

IVACB(2) = input value of the specific impulse at vacuum conditions for booster

I_{V_o} = input value of the vacuum specific impulse of the orbiter

= IVACO(5)

With a reasonably good input estimate for IDVEL (e.g., within 250 ft/sec), the basic synthesis iteration loop usually converges within three iterations. The test specified in Step 5 is performed in the subroutine ITER8 and the error is compared with the input tolerance on μ_o, TOLMU (≈.005 is usually acceptable). The estimates for the gross stage weights (W_{G_B}, W_{G_O}) need not be good (±50% is acceptable on vehicle size), since the convergence is very rapid and stable. The basic sizing constraints referenced in Step 3c above are discussed in Section 2.3.2 Basic Synthesis Options. Calculation of the booster cruise requirements and how the requirements interface with the basic synthesis iteration process are discussed in Section 2.3.4, Booster Subsonic Cruise Options. The basic synthesis iteration loop is continued until convergence or until the allowable number of iterations have been exceeded as specified by the input SYNIT (=6 as a maximum).

2.3.1.1 Weight Sizing Driver. The basic data bank of sizing laws are contained in the WTSCH subroutine. These sizing laws and their input requirements are described in detail in Volume II of this report, Weight/Volume Handbook. The basic input requirements to WTSCH are the scaling coefficients, fixed weights and estimates made to initiate the sizing process.

The basic scaling process performed by WTSCH results in component weights and volumes for the stage that are based on the estimate for the overall or gross stage weight and volume. Therefore an iteration process is employed to ensure that these two are mutually consistent. The iteration to ensure that the sum of the component volumes is mutually consistent with gross volume is contained within WTSCH and is a simple repetitive iteration on the total stage body volume. The iteration to ensure that the sum of the component weights is consistent with the gross stage weight (W_G) is the responsibility of the subroutine SOLVE. In SOLVE a Newton-Raphson iteration procedure is used to drive the difference in the gross stage weight that is the estimated input, W'_G, and the gross stage weight that is obtained from WTSCH, W_G, to zero, viz., $W_G - W'_G \rightarrow 0$. The Newton-Raphson iteration for $(n+1)^{th}$ step is as follows:

$$W'_{G_{(n+1)}} = W'_{G_{(n)}} - e_{(n)} \Bigg/ \left[\frac{e_{(n)} - e_{(n-1)}}{W'_{G_{(n)}} - W'_{G_{(n-1)}}}\right]$$

where, $W'_{G_{(n+1)}}$, $W'_{G_{(n)}}$, $W'_{G_{(n-1)}}$ = estimate for the stage gross weight for the $(n+1)^{th}$, $(n)^{th}$, and $(n-1)^{th}$ iterations, respectively.

$e_{(n)}$, $e_{(n-1)}$ = the error in the output vs. input of the gross weight, $W_G - W'_G$, for the $(n)^{th}$ and $(n-1)^{th}$ iterations, respectively.

This iteration is continued until the error for the $(n+1)^{th}$ step, $e_{(n+1)}$, is within specified input tolerance ($\approx$5 lbs) for each stage. The orbiter sizing is performed in this fashion with fixed payload, μ_O and vacuum thrust or vacuum thrust/gross weight. The orbiter gross weight obtained is set equal to the booster payload and the process is repeated for the booster with fixed μ_B and vacuum thrust or vacuum thrust/booster gross weight. Thus the data bank of sizing equations contained in WTSCH is used for both the orbiter and booster stages with the fixed weights, scaling coefficients, initial estimates, and sizing tolerance being input separately for each stage (see Section 4 of this volume and the Weight/Volume Handbook, Volume II of this report).

The basic output of this weight sizing iteration is a set of vehicle weights and component weights consistent with the constant velocity requirement imposed on the system. Optional vehicle constraints imposed on the system which require additional resizing of the configuration before updating this velocity requirement (i.e., simulating the ascent trajectory) are discussed in Section 2.3.2, Basic Synthesis Options.

2.3.1.2 Trajectory Simulation Driver (Ascent). Simulation of the ascent trajectory basically tests the vehicle configuration obtained from the WTVOL subprogram to insure that it meets the total velocity requirement which formed the basis of its sizing. In addition the output from the simulation forms the bases for a detailed analysis of the design trajectories for the shuttle configuration while providing an excellent means for determining sensitivities to the basic design due to changes in mission requirements. Simulation of the booster entry trajectory is provided as an option, however the only result obtained which is necessary to the synthesis process is the cruise range requirement which is used to estimate the booster cruise fuel requirements, in general a second order effect on the total system sizing. This option along with the calculation of the booster cruise fuel requirements is described in Section 2.3.4, Booster Subsonic Cruise Options.

Due to the complex interrelationship between the design of the shuttle and its mission, a baseline ascent trajectory profile was established and built into the basic synthesis process of the SSSP thus simplifying usage of the SSSP and providing a common basis for predesign studies. The subroutine TRAJC provides the basic logic that presets the input parameters to the GTSM subprogram which establishes this trajectory profile. The preset values of these parameters are described in Section 4, Program Operating Instructions.

Seven basic flight simulation sections are available to model the ascent flight profile. The following is a summary of the seven simulation sections to be used as a reference for subsequent discussion:

1. Booster ignition (instantaneous startup of main engines)
   a. vertical rise for an input specified time ($\approx$8-10 seconds)
   b. pitchover maneuver at a constant relative pitch rate for an input specified time ($\approx$16 seconds)
   c. gravity turn maneuver (pitch angle-of-attack = 0) to a section terminal condition specified by input (usually a time point or the point at initiation of main engine throttling if required)
2. Gravity turn maneuver to boost-phase propellant depletion (instantaneous shutdown of main engines)
3. Coast of the attached vehicles for an input specified time. Termination of this simulation section is considered the point at which the vehicles are staged or separated and the point at which the booster entry/return is initiated.
4. Coast of the orbiter stage for an input specified time.
5. Orbiter ignition (instantaneous startup of main engines) and burn for an input specified time.
6. Orbiter burn to a section terminal condition specified by input (usually a time point or the point at initiation of main engine throttling if required).

7. Orbiter burn to orbit insertion, i.e., attainment of the input specified inertial velocity (V_f).

The vehicle attitude during the short coast periods in flight phases 3 and 4 above may be modeled as either a gravity turn or a specified constant pitch rate maneuver as specified by input. A multiplier on the initial pitch rate during flight phase 1 above is iteratively determined to yield an input specified relative flight path angle(or Q) at termination of flight phase 3 above. The orbiter flight (phases 5, 6 and 7 above) may be simulated with the cotangent of the pitch attitude being linear in time, $\cot\psi = A+Bt$. The two parameters A and B are iteratively determined to yield the input specified injection attitude (h_f) and injection flight path angle (γ_f) at attainment of the input specified injection velocity (V_f). (An optional method of specifying the ascent trajectory terminal conditions is discussed in Section 2.2.2, Trajectory Computation, where the injection point is input specified by the perigee and apogee atitudes of the parking orbit and the true anomaly of the orbit at the insertion point.) All of the specified input parameters mentioned above is input to the GTSM data block ($DATA1) discussed in Section 4 of this report.

The basic output derived from the ascent trajectory simulation is the orbiter propellant required to achieve V_f thus determining the orbiter main impulse mass ratio, μ_o^*. In addition the maximum value of the dynamic pressure (Q_{MAX}) obtained during the ascent is stored for use in the sizing equations in WTSCH for the next basic synthesis iteration. The separation condition flight parameters (at termination of flight phase 3 above) are also stored for use in determining the booster cruise range requirements and update of the booster cruise fuel requirements for use during the next basic synthesis iteration. These updating processes utilized for these parameters are discussed in the following sections.

The basic output from the WTVOL subprogram necessary to simulate the baseline ascent trajectory is prepared in the subroutine TAMPER prior to entry of the GTSM subprogram. These output parameters are obtained directly from the last pass through the WTSCH subroutine for each stage and include the following computed items:

W_{G_B} = gross weight of the booster

W_{G_O} = gross weight of the orbiter

W_{P_B} = booster main propellant expended during ascent

$= W_{G_B} * (\mu_B - 1)/\mu_B$

S_{W_B}, S_{W_O} = total (theoretical) wing areas of the booster and orbiter

T_{V_B}, T_{V_O} = total stage vacuum thrust of the booster and orbiter

The booster gross weight is set equal to the initial weight at the start of flight phase 1 above. If fixed-size solid rockets are utilized for thrust augmentation at liftoff, their gross weight is added to W_{G_B} to form the total vehicle weight at liftoff. In addition flight phase 1 is automatically terminated on the burn time of the solid rockets at which time the spent rockets are jettisoned. This option is discussed in Section 2.3.3, Secondary Synthesis Options. The orbiter gross weight is set equal to the initial weight at the start of flight phase 4 above. The booster propellant expended is used to determine the section terminal condition of flight phase 2. Use of the orbiter main engines (fed from the orbiter propellant tanks) during the initial boost phase is also discussed in Section 2.3.3, Secondary Synthesis Options. Use of this option modifies the section terminal condition to include this propellant expenditure. The theoretical wing areas are utilized as the aerodynamic reference areas for computation of the aerodynamic forces on the vehicle during the trajectory simulation. This process performs the scaling of the aerodynamic forces on the vehicle as the vehicle size and wing size varies during the synthesis iteration process. (NOTE: the aerodynamic force coefficients remain unchanged and are fixed specified input and should typify the shuttle configuration being examined.) The propulsion characteristics are defined for each ascent flight phase and utilize T_{V_B} and T_{V_O} computed from WTSCH, input specified specific impulses (at sea level and vacuum conditions) for each stage and input multiplicative thrust factors which are used to control variations in engine performance during flight such as utilization of emergency power levels or changing the nozzle expansion ratio. (These factors are also used to control certain propulsion options, such as simultaneous burn of both stages at liftoff.)

The subroutine TAMPER supplies the GTSM subprogram with the total vacuum thrust (T_V) and effective specific impulse at vacuum and sea level conditions, I_V and I_{SL}, are defined for each of the flight phases as follows:

1. $$T_V = T_{V_B} * TFCTRB(1) + T_{V_O} * TFCTRO(1)$$

$$I_V = \frac{T_V}{\dot{W}}$$

$$I_{SL} = \frac{1}{\dot{W}} \left[T_{V_B} * \frac{ISLB(1)}{IVACB(1)} * TFCTRB(1) + T_{V_O} * \frac{ISLO(1)}{IVACO(1)} * TFCTRO(1) \right]$$

where

$$\dot{W} = T_{V_B} * \frac{TFCTRB(1)}{IVACB(1)} + T_{V_O} * \frac{TFCTRO(1)}{IVACO(1)}$$

2. T_V, I_V, and I_{SL} are computed the same as in flight phase 1 with the replacements in the equations:

TFCTRB(1)	=	TFCTRB(2)
TFCTRO(1)	=	TFCTRO(2)
IVACB(1)	=	IVACB(2)
ISLB(1)	=	ISLB(2)
IVACO(1)	=	IVACO(2)
ISLO(1)	=	ISLO(2)

3.&4. Coast: T_V, I_V and I_{SL} are set to zero internally. (NOTE: the subscripted parameters listed above are also input specified but utilized for printout purposes only where the subscript "3" is considered the nominal condition and the subscript "4" is considered the uprated condition of the parameter.)

5,6 & 7.

$$T_V = T_{V_O} * \text{TFCTRO(I)}$$

$$I_V = \text{IVACO(I)}$$

$$I_{SL} = \text{ISLO(I)}$$

where I = 5, 6 or 7

The thrust factors may be used to show the effect on system sizing or performance * by uprating the main engine performance or by using the orbiter engines during the boost phase. Use of the thrust factors for the orbiter during flight phases 1 or 2 require that the input value of FIRE be set equal to 1., however (see Section 2.3.3, Secondary Synthesis Options). The normal operational procedure for the shuttle usually requires only sequential burns of the stages and therefore would not require these thrust factors for the orbiter and hence the input value of FIRE must be set equal to 2., see Section 2.3.3.1, Sequential Stage Burns.

*Propellants expended only

2.3.2 BASIC SYNTHESIS OPTIONS

The basic options provided by the SSSP operate within the WTVOL subprogram prior to entry into the GTSM subprogram. In this manner the vehicle configuration is sized within specified constraint tolerances with constant total velocity requirements. These basic options include constraints due to commonality between the engine sizes of both stages or due to restrictions placed on the gross weights of the stages. The first two options describe the basic propulsion system procedures one of which must be selected to model the configuration. In additon, the sequencing of the stage burns must be selected, i.e., the orbiter main engines may or may not be utilized during the booster burn to staging. Sequencing of stage burns is discussed in the next section, Secondary Synthesis Options, Section 2.3.3, since this procedure does not directly enter into the synthesis process. The last three options in this section describe restrictions which may be placed on the gross size of the stages however, they need not be utilized to examine system performance.

2.3.2.1 Fixed Booster Thrust (Common Stage Engines). This option specifies that the thrust level per engine (or unit thrust) for the booster is obtained from the total vacuum thrust level of the orbiter, T_{V_O}. The vacuum unit thrust or the vacuum thrust/gross weight of the orbiter is fixed input specified in the orbiter input data block (see Volume II, Weight/Volume Handbook). The T_{V_O} is thus obtained from the orbiter sizing (WTSCH). Prior to the booster sizing entry to WTSCH, the booster thrust/weight factor is set to zero and the vacuum unit thrust of the booster, T_{V_B}/N_B, is computed as follows:

$$T_{V_B}/N_B = T_{V_O} * \left[\frac{TRATIO}{N_O} \right]$$

where N_B and N_O are the fixed number of booster and orbiter engines as specified in the booster and orbiter input data blocks respectively, and TRATIO is the input specified scaling factor for the ratio of the booster vacuum unit thrust to orbiter vacuum unit thrust. This factor is input ($\approx$1) to account for the differences in nozzle expansion ratios between the stages. The booster is thus sized on this vacuum unit thrust yielding the gross stage weight, W_{G_B}.

The following input flags must also be specified to obtain this option:

BOOTW = 0.,

TWLOI = -1.,

Use of this option includes an important initial check on the basis of the sizing process prior to either simulating the ascent trajectory of the vehicle or testing for additional constraints placed on the configuration. This check is performed to ensure the initially-sized vehicle has been synthesized with a set of estimates that yields a reasonable thrust/weight at liftoff for the configuration (>1.2)*. The liftoff thrust/weight, $T/W)_{LO}$, is computed in the subroutine TAMPER utilizing W_{G_B} and the actual liftoff thrust for flight phase 1 as discussed in Section 2.3.1.2 above. Adjustments are made to the μ_o and the estimate for the maximum dynamic pressure, Q_{MAX}, utilized to size certain aerodynamic control surfaces on each stage (see Volume II, Weight/Volume Handbook), to anticipate the effect of changes in $T/W)_{LO}$. The vehicle is then resized with the adjusted values. If the process results in a "poor" $T/W)_{LO}$ after two iterations, the run is aborted which implies that either a larger number of booster engines, N_B, are required or a higher orbiter thrust/weight or orbiter unit thrust is needed. The adjustments made to μ_o and Q_{MAX} are based on sensitivities derived from past studies of typical shuttle configurations:

*This check is not made if GWREQ >0 (see Section 2.3.2.3, Fixed Booster Gross Weight) and with the vacuum unit thrust of the orbiter specified since this assures that the liftoff thrust/weight is fixed and known, a priori.

$$\mu_{o_{(n+1)}} = \mu_{o_{(n)}} * \left[1 + \frac{T/W)_{LO_{(n)}} - T/W)_{LO_{(n-1)}}}{(1 - X_{(n)}) \left[T/W)_{LO_{(n)}} - 1 \right]} \right]^{Y}$$

and,

$$Q_{MAX_{(n+1)}} = Q_{MAX_{(n)}} + QMXS * \frac{T/W)_{LO_{(n)}} - T/W)_{LO_{(n-1)}}}{(1 - X_{(n)})}$$

where,

$$X_{(n)} = \mu_{o_{(n)}} * \frac{500}{32.174 * IVACO(7) * \left[T/W)_{LO_{(n)}} - 1 \right]} - 0.2$$

$$Y = \frac{-1000}{32.174 * IVACO(7)}$$

IVACO(7) = input specified vacuum specific impulse for the orbiter

QMXS = input specified estimate for the slope of Q_{MAX} vs. $T/W)_{LO}$: Input $\approx$ 900.

The subscript n implies the n[th] adjustment and is equal to 2 or 3 (two cycles are allowed). The first sizing pass is based on the initial computed estimate for μ_o with IDVEL (Section 2.3.1, Basic Synthesis Iteration) and an input specified estimate for Q_{MAX} = QMAX.

2.3.2.2 Fixed Liftoff Thrust-to-Weight. (a) Non-Common Stage Engines. This option specifies that the vehicle thrust-to-weight at liftoff be held constant independent of the gross liftoff weight obtained from the sizing process (booster output from WTSCH) and the thrust level specified for the orbiter. The vacuum unit thrust or the vacuum thrust/gross weight of the orbiter is fixed specified input in the orbiter data block (see Volume II, Weight/Volume Handbook). The thrust/weight input to the booster data block is assumed to be the required value at liftoff (sea level conditions). Prior to the booster sizing entry to subroutine WTSCH, the booster vacuum unit thrust is set to zero and the booster thrust/weight factor at vacuum conditions is computed as follows:

$$T/W)_B = T/W)_{LO} * \frac{IVACB(1)}{ISLB(1)}$$

where,

$T/W)_B$ = vacuum thrust/gross weight of the booster required to scale the booster in subroutine WTSCH

$T/W)_{LO}$ = required liftoff thrust/weight input to the booster data block

IVACB(1) = input specified booster vacuum specific impulse

ISLB(1) = input specified booster sea level specific impulse

The following input flag must also be specified to obtain this option: BOOTW=1.,

(b) Common Stage Engines. This option specifies that the vehicle thrust-to-weight at liftoff be held constant independent of the gross liftoff weight obtained from the sizing process (booster output from subroutine WTSCH) and so that the following relationship exists between the booster and orbiter thrust levels during the WTSCH sizing process:

$$T_{V_B}/N_B = T_{V_O} * \left[\frac{TRATIO}{N_O} \right]$$

where,

T_{V_B}/N_B = vacuum unit thrust of the booster

T_{V_O} = total vacuum thrust of the orbiter

N_O = number of orbiter engines

TRATIO = input specified scaling factor (see option described in Section 2.3.2.1 above)

Since the orbiter stage is sized prior to sizing the booster, the orbiter thrust/weight factor is iteratively determined so that the computed liftoff thrust-to-weight for the vehicle is within input specified tolerance of the desired value. Thus the input specified value of the orbiter vacuum thrust/gross weight becomes the initial estimate for this parameter and the vacuum unit thrust for the orbiter is internally set to zero. After the orbiter has been sized, the booster is sized with the vacuum unit thrust computed from the above relationship yielding the gross weight of the booster stage, W_{G_B}. The total vehicle thrust/weight at liftoff, $T/W)_{LO}$, is then computed in subroutine TAMPER as follows, using the input specified values of the specific impulses and thrust factors for flight phase 1:

$$T/W)_{LO} = \frac{1}{W_{G_B}} * \left[T_{V_B} * \text{TFCTRB(1)} * \frac{\text{ISLB(1)}}{\text{IVACB(1)}} \right.$$

$$\left. + T_{V_O} * \text{TFCTRO(1)} * \frac{\text{ISLO(1)}}{\text{IVACO(1)}} \right]$$

Use of the factors on T_{V_O} in this equation are discussed in Section 2.3.3, Secondary Synthesis Option(with FIRE=1 and optional use of the orbiter main engines at liftoff for thrust augmentation)along with modifications to this equation if fixed solid rocket motors are used at liftoff for thrust augmentation. This computed value of the vehicle liftoff thrust/weight is then compared with the desired specified value input by TWLO. If the difference is greater than the allowed as specified by the input value of the tolerance, TOLTW ($\approx$.001) the following correction is made to the $(n+1)^{th}$ iteration estimate of the orbiter vacuum thrust/gross weight ratio, $T/W)_O$:

$$T/W)_{O_{(n+1)}} = T/W)_{O_{(n)}} * \frac{\text{TWLO}}{T/W)_{LO_{(n)}}}$$

where

$T/W)_{o_{(n+1)}}$, $T/W)_{o_{(n)}}$ = $(n+1)^{th}$ and $(n)^{th}$ iteration estimate value of the orbiter vacuum thrust/gross weight

TWLO = input specified thrust/weight for the vehicle at liftoff

$T/W)_{LO_{(n)}}$ = computed value of the liftoff thrust/weight for the $(n)^{th}$ iteration.

This iteration process is continued until convergence or until the input iteration counter, TWLOI ($\approx$6), is exceeded.

The following input flags must be specified to obtain this option:

BOOTW	= 0.,
TWLO	= desired vehicle liftoff thrust/weight ratio
TWLOI	= number of maximum allowable iterations to obtain TWLO $\approx$6.
TOLTW	= tolerance on TWLO $\approx$.001

2.3.2.3 Fixed Booster Gross Weight (Liftoff Weight). The normal procedure utilized in the WTVOL subprogram provides for sizing the shuttle configuration for a fixed payload as input - specified by the orbiter stage input data block (see Volume II, Weight/Volume Handbook), thus obtaining the gross weight of the orbiter and booster, respectively. The gross weight of the orbiter automatically becomes the booster payload. The booster gross weight is basically considered the gross liftoff weight of the system.

This option provides for the capability of fixing the booster gross weight by specifying the required value with the input GWREQ and solving for the orbiter payload. Since the orbiter stage is sized prior to sizing the booster, the payload is iteratively determined so that the computed gross weight of the booster, W_{G_B}, is within tolerance of the desired gross weight. (NOTE: if fixed solid rocket motors are used for thrust augmentation during boost, the gross weight of the solids should not be included in GWREQ since W_{G_B} is not sized with them; see Section 2.3.3, Secondary Synthesis Options).

A Newton-Raphson iteration procedure is used to solve for the payload, PL, that yields the desired booster gross weight, GWREQ:

$$PL_{(n+1)} = PL_{(n)} - (GWREQ - W_{G_{B(n)}}) \Bigg/ \left[\frac{W_{G_{B(n-1)}} - W_{G_{B(n)}}}{PL_{(n)} - PL_{(n-1)}} \right]$$

where,

$PL_{(n+1)}, PL_{(n)}, PL_{(n-1)}$	=	$(n+1)^{th}$, n^{th}, and $(n-1)^{th}$ iteration estimate of the value of the system payload, respectively
GWREQ	=	input specified booster gross weight
$W_{G_{B(n)}}$, $W_{G_{B(n-1)}}$	=	computed value of the booster gross weight for the $(n)^{th}$ and $(n-1)^{th}$ iteration, respectively.

This iteration process is continued until convergence or until the maximum allowable iterations have been exceeded (internally set at 5). Convergence is usually very rapid and is obtained within 3 or 4 iterations. The built-in tolerance on the iteration process requires that the computed value of the W_{G_B} is within .005% of the desired value, GWREQ. For typical configurations this yields a gross weight that is within ± 250 pounds of the desired value.

The input value of GWREQ is also the input flag that specifies use of this option. If GWREQ ≤ 0 this option is automatically bypassed. In addition this option should only be used in conjunction with the Fixed Booster Thrust option described in Section 2.3.2.1 above (BOOTW = 0., and TWLOI = -1.,) where the vacuum unit thrust level of the orbiter is input specified.

2.3.2.4 Fixed Orbiter Gross Weight. This option provides for the capability of fixing the orbiter gross weight by specifying the required value with the input WOREQ. As mentioned earlier the normal procedure utilized in the WTVOL subprogram provides for sizing the shuttle configuration based on a constant total characteristic velocity requirement, ΔV, which is the sum of the incremental characteristic velocities required of the booster and orbiter stages. In WTVOL the total characteristic velocity is considered to be approximately proportional to the product of the mass ratios, $\mu_O * \mu_B$ = constant = $\exp(\Delta V/Ig)$, where I may be considered the overall system specific impulse. The computed gross weight of the orbiter, W_{G_O}, is then implicitly defined in terms of the booster mass ratio (μ_B),

$$W_{G_O} = f(\mu_o) = f\left[\exp(\Delta V/Ig) / \mu_B\right].$$

Since the orbiter stage is sized prior to sizing the booster stage, the desired value of W_{G_O} is iteratively determined by varying μ_B within the process. Hence this process essentially varies the velocity distribution between the stages.

A Newton-Raphson iteration procedure is used to solve for μ_B that yields the desired orbiter gross weight, WOREQ:

$$\mu_{B_{(n+1)}} = \mu_{B_{(n)}} - \left(WOREQ - W_{G_{O_{(n)}}}\right) \Bigg/ \left[\frac{W_{G_{O_{(n-1)}}} - W_{G_{O_{(n)}}}}{\mu_{B_{(n)}} - \mu_{B_{(n-1)}}}\right]$$

where,

$\mu_{B_{(n+1)}}, \mu_{B_{(n)}}, \mu_{B_{(n-1)}}$ = $(n+1)^{th}$, $(n)^{th}$, and $(n-1)^{th}$ iteration estimate of the booster mass ratio, respectively. The first estimate of μ_B is input through the booster input data block (see Section 4 of this volume and Volume II, Weight/Volume Handbook)

WOREQ = input specified orbiter gross weight

$W_{G_{O_{(n)}}}, W_{G_{O_{(n-1)}}}$ = computed value of the orbiter gross weight for the $(n)^{th}$ and $(n-1)^{th}$ iteration, respectively.

This iteration process is continued until convergence or until the maximum allowable iterations have been exceeded (internally set at 6). Convergence is usually very rapid and is obtained within 3 or 4 iterations. The built-in tolerance on the iteration process requires that the computed value of W_{G_O} is within .025% of the desired value, WOREQ. For typical configurations this yields a gross weight that is within ± 250 pounds of the desired value.

The input value of WOREQ is also the input flag that specifies use of this option. If WOREQ ≤ 0 this option is automatically by-passed. This option may be used in conjunction with any one of the basic synthesis options described in the preceding paragraphs, but not with the Fixed Orbiter Propellant Weight option described in the next section, Section 2.3.2.5. NOTE: if this option is not used, the system is sized with a fixed value for μ_B as input specified in the booster input data block.

Use of this option includes an adjustment to μ_O during the iteration process described above. This adjustment not only improves convergence of this option but also improves the Basic Synthesis Iteration process (described in Section 2.3.1) by anticipating the effect varying the velocity distribution between the stages by changing μ_B. During each Newton-Raphson iteration above, the following adjustment to μ_O is made:

$$\mu_{O_{(n+1)}} = \mu_{O_{(n)}} * \frac{\mu_{B_{(n)}}}{\mu_{B_{(n+1)}}}$$

where,

$\mu_{O_{(n+1)}}$, $\mu_{O_{(n)}}$ = $(n+1)^{th}$ and $(n)^{th}$ adjustment to the orbiter mass ratio used in sizing during the $(n+1)^{th}$ and $(n)^{th}$ iteration on the booster mass ratio, respectively

$\mu_{B_{(n+1)}}$, $\mu_{B_{(n)}}$ = $(n+1)^{th}$ and $(n)^{th}$ iteration estimate of the booster mass ratio to obtain the desired orbiter gross weight, respectively.

2.3.2.5 Fixed Orbiter Propellant Weight. This option provides for the capability of fixing the total "impulse" propellant expended during the orbiter burn (s) by specifying the required value with the input WPOREQ. The orbiter total "impulse" propellant is defined as the sum of the ascent main-propellants burned to parking orbit insertion (excluding flight performance reserves, trapped propellants, losses, etc.) and the total post-insertion (or on-orbit) maneuvering propellants required, if any, (See Secondary Propellant Weights, Section 13.1.8 of Volume II, Weight/Volume Handbook). Defining the orbiter total impulse propellants in this fashion allows the user to readily specify the actual volume of propellants utilized in meeting the total mission velocity requirements.

The iterative procedure and methodology utilized for this option is identical to that used in the Fixed Orbiter Gross Weight option described in Section 2.3.2.4 since the orbiter sizing is performed prior to the booster synthesis and the orbiter total impulse propellants, W_{P_O}, are also implicitly defined as a function of the booster mass ratio, μ_B. Therefore the desired value of the orbiter total impulse propellants is iteratively determined by varying μ_B within the process.

A Newton-Raphson iteration procedure is used to solve for μ_B that yields the desired value of the orbiter propellants, WPOREQ:

$$\mu_{B_{(n+1)}} = \mu_{B_{(n)}} - \left(\text{WPOREQ} - W_{P_{O_{(n)}}}\right) \Bigg/ \left[\frac{W_{P_{O_{(n-1)}}} - W_{P_{O_{(n)}}}}{\mu_{B_{(n)}} - \mu_{B_{(n-1)}}}\right]$$

where,

$\mu_{B_{(n+1)}}, \mu_{B_{(n)}}, \mu_{B_{(n-1)}}$ = $(n+1)^{th}$, $(n)^{th}$, and $(n-1)^{th}$ iteration estimate of the booster mass ratio, respectively. The first estimate of μ_B is input through the booster input data block (see Section 4 of this volume and Volume II, Weight/Volume Handbook)

WPOREQ = input specified orbiter total impulse propellant

$W_{P_{O_{(n)}}}, W_{P_{O_{(n-1)}}}$ = computed value of the orbiter total impulse propellant for the $(n)^{th}$ and $(n-1)^{th}$ iteration, respectively.

This iteration process is continued until convergence or until the maximum allowable iterations have been exceeded (internally set at 6). Convergence is usually very rapid and is obtained within 3 or 4 iterations. A built-in tolerance on the iteration process yields a computed value of W_{P_O} within .025% of WPOREQ or about ± 200 pounds of the absolute value of WPOREQ.

The input value of WPOREQ is also the input flag that specifies use of this option. If WPOREQ ≤0 this option is automatically by-passed. This option may be used in conjunction with any of the Basic Synthesis Options described in this section except with the Fixed Orbiter Gross Weight option (Section 2.3.2.4) and therefore, $WOREQ \leq 0$. NOTE: if this option is not used, the system is sized with a fixed value for μ_B as input specified in the booster input data block.

Use of this option also includes an adjustment to μ_O during the iteration procedure and is identical to the scheme described in the Fixed Orbiter Gross Weight Option.

2.3.3 SECONDARY SYNTHESIS OPTIONS

The secondary synthesis options provided by the SSSP are so-called because they were developed for use in preliminary studies during the early phases of configuring the Space Shuttle concept. These options basically provide for thrust augmentation during the boost phase of flight (flight phases 1 or 2)* to minimize the number or size of the main rocket engines required for the booster stage. As with the Basic Synthesis Options (Section 2.3.2), the secondary options operate within the WTVOL subprogram prior to entry into the GTSM subprogram and simulation of the ascent trajectory. The first option discusses the normal operational procedure for the shuttle concept and the SSSP.

2.3.3.1 Sequential Stage Burns. The normal operational procedure utilized for the ascent flight of the shuttle involves sequential burn of the booster and orbiter stages to parking orbit insertion. The booster main rocket engines are ignited at liftoff and are burned to propellant depletion. Staging and separation of the stages occur a few seconds later. The orbiter main rocket engines are then ignited and the orbiter stage continues to parking orbit insertion. This procedure is obtained in the SSSP by specifying the input FIRE=2., selecting one of the basic propulsion procedures discussed in Section 2.3.2 above to model the configuration and then supplying the input specified constant specific impulses and thrust factors for each flight phase as described in the Trajectory Simulation Driver (Ascent), Section 2.3.1.2. In this mode of operation the values of the orbiter thrust factors during

*See Trajectory Simulation Driver (Ascent), Section 2.3.1.2

flight phases 1 and 2 are ignored and need not be input. The remainder of the thrust factors may be utilized to account for variations in engine performance during flight such as utilization of emergency power levels or changing the nozzle expansion ratio with an extendable skirt. They have nominal built-in values equal to 1.

2.3.3.2 Simultaneous Stage Burns. This optional procedure allows for utilizing the orbiter stage main rocket engines during the initial boost phase of flight (flight phases 1 and 2 as described in Section 2.3.1.2, Trajectory Simulation Driver). Two methods for accommodating this procedure are described in the following paragraphs.

a. No Cross feed. This option allows the orbiter main rocket engines to be utilized during the initial boost phase for thrust augmentation. Use of the orbiter engines in this mode requires use of the propellants from the orbiter tanks during the boost burn thereby depleting the supply of propellants available to the orbiter during its solo burn to orbit insertion (i.e., no crossfeed of propellants from the booster tanks to the orbiter engines). Usually the burn of the orbiter engines in this mode is of short duration and assumes that the engines are in-flight restartable or have a wide range of throttling capability since use of the orbiter main propellants for this purpose has proven to be rather expensive in terms of overall system performance.

This procedure is obtained by specifying the following input:

FIRE=1.,

NXFOB=1., (or $\neq$ 0.,)

selecting one of the basic propulsion procedures discussed in Section 2.3.2 above to model the configuration, and then supplying the input specified constant specific impulses and thrust factors for each flight phase as described in the Trajectory Simulation Driver (Ascent) Section 2.3.1.2. The thrust factors may be utilized to account for variations in engine performance during flight however, these factors have nominal built-in values equal to 1. In addition the terminal condition of flight phase 1 must be input specified as a time point after liftoff (see Section 4 for trajectory simulation control) which is used in the subroutine TAMPER to compute the total orbiter propellant expended during flight phases 1 and 2 in this mode and hence, the automatic termination value of flight phase 2, boost-phase propellant depletion.

b. With Crossfeed. This option allows the orbiter main rocket engines to be utilized during the initial boost phase for thrust augmentation without depleting the propellants in the orbiter tanks. Use of the orbiter engines in this mode requires use of the booster main propellants in a crossfeed manner to the orbiter engines thereby depleting the supply of propellants available to the booster during ascent. NOTE: the weight of the propellant feed lines and necessary hardware items to accommodate this option should also be accounted for in the input data blocks of either the booster or orbiter stages ($DATA3, see Section 4 of this volume).

This procedure is obtained by specifying the following input:

FIRE =1.,

NXFOB =0.,

selecting one of the basic propulsion procedures discussed in Section 2.3.2 above to model the configuration, and then supplying the input specified constant specific impulses and thrust factors for each flight phase as described in the Trajectory Simulation Driver (Ascent), Section 2.3.1.2. The thrust factors may be utilized to account for variations in engine performance during flight however, these factors have nominal built-in values equal to 1.

2.3.3.3 Fixed Solid Rocket Motor Augmentation. This option provides for thrust augmentation during the initial boost phase of flight by utilizing fixed solid rocket strap-ons. The characteristics of the solids are fixed input specified and are modeled in a very simplified manner. Use of the solids is limited to flight phase 1 described in Section 2.3.1.2, Trajectory Simulation Driver (Ascent).

The vacuum thrust time history for the solid rocket(s) is assumed to be a linear function of time from ignition:

$$T_{V_{SRM}} = SOLID * (AS + BS * t)$$

where,

$T_{V_{SRM}}$	=	total solid rocket vacuum thrust
SOLID	=	input specified number of solid rockets
AS	=	input specified constant (single rocket vacuum thrust at t=0)
BS	=	input specified constant slope
t	=	time (seconds) from ignition of solids

The modeling of the solid rocket thrust during simulation of flight phase 1 accounts for the atmospheric pressure on the nozzle exit areas as follows:

$$T_{SRM} = T_{V_{SRM}} - SOLID * p_a * SAE$$

where,

T_{SRM}	=	total solid rocket thrust
p_a	=	atmospheric pressure (psi) = f (altitude)
SAE	=	input specified nozzle exit area (in^2) per rocket

The total solid rocket propellant flow rate, $\dot{W}_{SRM}$, is computed as:

$$\dot{W}_{SRM} = T_{V_{SRM}} / SISP$$

where SISP = input specified constant specific impulse (sec) for the solid rocket(s). The total solid rocket propellant expended, $W_{P_{SRM}}$, is computed as:

$$W_{P_{SRM}} = \frac{SOLID}{SISP} * TSBO * (AS + 0.5 * BS * TSBO)$$

where TSBO = input specified burn time for solid rockets. The gross weight for solid rockets, $W_{G_{SRM}}$, is computed as:

$$W_{G_{SRM}} = W_{P_{SRM}} + SOLID * SINERT$$

where SINERT = input specified inert weight per solid rocket. The input value of TSBO is used to automatically terminate flight phase 1 at which time the spent solid rockets, $W_{j_{SRM}}$, are jettisoned where

$$W_{j_{SRM}} = SOLID * SINERT.$$

The input value of SOLID is also the input flag that specifies use of this option. If SOLID ≤ 0 this option is automatically by-passed. This option may be used in conjunction with any of the Basic Synthesis Options (Section 2.3.2) or secondary options described above with the following exception: Fixed Liftoff Thrust-to-Weight, Non-Common Stage Engines (Section 2.3.2.2a). This option may be used in conjunction with the Common Stage Engines part of this basic option (Section 2.3.2.2b) however, since the iteration procedure described computes the total vehicle thrust/weight at liftoff, $T/W)_{LO}$, which is compared to the desired value, TWLO, the following modification is made in the subroutine TAMPER to include the solid rockets:

$$T/W)_{LO} = \frac{\left[T_{V_B} * TFCTRB(1) * \frac{ISLB(1)}{IVACB(1)} + T_{V_O} * TFCTRO(1) * \frac{ISLO(1)}{IVACO(1)}\right]}{W_{G_B} + W_{G_{SRM}}}$$

$$+ T_{SL_{SRM}} \Big/ \left[W_{G_B} + W_{G_{SRM}}\right]$$

where $T_{SL_{SRM}}$ is the total solid rockets thrust at sea level conditions at ignition (t=0) computed as:

$$T_{SL_{SRM}} = SOLID * (AS - 14.69 * SAE)$$

NOTE: if this option is used in conjunction with the Fixed Booster Gross Weight option described in Section 2.3.2.3, the gross weight of the solid rockets, $W_{G_{SRM}}$, are not included in the iteration process and therefore should not be accounted for in the input value of GWREQ.

2.3.4 BOOSTER SUBSONIC CRUISE OPTIONS

Inherent in the basic design of the Space Shuttle system is the concept of reuseability of the stages. This section concerns itself specifically with recovery of the booster stage and how the SSSP accounts for the cruise fuel allotment necessary to power the airbreathing engines for successful return to a suitable landing site or airport.

During the Basic Synthesis Iteration procedure described in Section 2.3.1, the booster is sized during each basic iteration in the WTVOL subprogram with the gross weight of the orbiter, W_{G_O}, as its payload, a fixed main-impulse mass ratio, μ_B, and a fixed total thrust or thrust/weight ratio. The result of this synthesis is primarily the gross weight of the booster, W_{G_B}, which describes the overall system performance. An important input parameter to the booster synthesis process is the allotment of the airbreathing fuel requirements. This parameter is also updated during each basic iteration and is provided for in terms of the performance mass ratio required for the cruise portion of flight (see Section 13.1.3 of Volume II, Weight/Volume Handbook). The cruise performance mass ratio is derived from Breguet's endurance equations for an aircraft cruising for a given distance at a constant speed, lift/drag ratio, and constant specific fuel consumption for the airbreathing engines. These constants are input specified. The distance or range that the booster must cruise after entry and transition to subsonic flight is supplied by various options within the SSSP. These options are primarily based on the flight conditions at the staging point supplied from ascent trajectory simulation. This data becomes the initial conditions for hypersonic entry trajectory and transition flight to subsonic speeds. Thus this update procedure operates directly with both the WTVOL and GTSM subprograms.

This section first describes the various options available that provide the basic subsonic cruise range requirement for calculation of the booster cruise mass ratio. Finally three alternative procedures are outlined which may be used to calculate this performance mass ratio in the form utilized by the synthesis procedure in WTVOL. One procedure must be selected from each of the two following subsections to define the booster cruise requirements and hence to successfully drive this portion of the SSSP.

2.3.4.1 Booster Range Calculation. This section describes the five basic options provided by the SSSP in order to determine the booster subsonic cruise range requirements.

a. Parametric Flyback Range Data. This option provides for determining the cruise range requirement for flyback to the launch site with the use of built-in data generated parametrically as a function of the flight conditions at staging initial conditions for entry for a typical shuttle configuration. This data was generated external to the SSSP using a three-degree-of-freedom trajectory simulation for a range of initial conditions, and was then incorporated into the SSSP in the form of curve fits or as tables (subroutine RANGE). The staging conditions necessary to drive this data is supplied by the simulation of the ascent trajectory. The output provides the range increment from the staging point to initiation of the subsonic cruise which is in the downrange direction from the launch site. The range increment from the launch site to the staging point (also provided by the ascent trajectory simulation) is then added to form the flyback range required. Provision for a bias on this range requirement has also been made to account for variations from the basis of the parametric data results. The basis for generation of the parametric data is discussed in what follows.

The nominal booster recovery trajectories were initiated at the staging point and terminated at an altitude of 50,000 feet. The following hypersonic vehicle characteristics were used for the trajectories:

$$\text{Entry loading parameter, } W/C_{L_{max}} S = 80.7 \text{ psf}$$

$$L/D \text{ at } C_{L_{max}} = 0.5$$

$$C_{L_{max}} = 0.723$$

The vehicle pitch attitude was assumed to be at zero lift at staging and be changed instantaneously to the $C_{L_{max}}$ attitude at apogee and maintained in that attitude until termination. The pitch angle-of-attack at $C_{L_{max}}$ for the vehicle used was 60 degrees. The vehicle bank angle was used to minimize downrange travel within a maximum normal load factor constraint of 4.0 g. This bank angle was maintained at this constant preselected value after staging until terminal altitude was reached (approximate change in heading angle of 180 degrees). The range increment from the staging point to the terminal point was then computed as the great-circle distance between these points. The staging conditions which initiate this entry trajectory were parametrized, within reasonable bounds, to generate the compilation of range increments. The basic staging conditions utilized were the altitude (h_s), relative velocity (V_s), and relative flight path angle(γ_s). Table 2-9 summarizes the output from this procedure and illustrates the approximate range of the data. This data is compiled into the subroutine RANGE. RANGE is called to yield the appropriate range increment after the staging conditions are determined from the ascent trajectory simulation. The table look-up subroutine TBL2D is used to

Table 2-9. Parametric Range Increments from Staging to Flyback Initiation

Note: Range in nautical miles.

h_s (ft)	γ_s (deg)	V_s (ft/sec) 7000	8000	9000	10000	11000	12 000	13000
18000	2	93	108	122	138	153	169	184
	4	101	118	134	153	171	190	209
	6	110	130	149	173	196	225	256
	8	120	145	168	199	232	267	304
	10	132	161	190	228	272	331	404
	12	144	179	212	270	338	413	507
190000	2	99	116	133	150	168	185	204
	4	107	127	146	168	190	215	241
	6	117	140	163	192	222	252	285
	8	127	151	183	219	252	298	354
	10	139	171	203	253	312	378	455
	12	152	189	237	303	372	457	-534
200000	2	105	123	143	165	187	209	230
	4	113	135	158	186	212	239	266
	6	122	148	177	210	241	271	310
	8	133	163	195	237	284	339	403
	10	146	180	224	284	345	411	496
	12	159	203	260	329	403	-480	-559
210000	2	111	130	152	176	200	224	248
	4	120	142	167	196	225	253	283
	6	131	154	182	219	260	302	349
	8	142	169	206	259	316	375	441
	10	153	188	240	304	371	446	-530
	12	168	213	270	347	-432	-497	-573
220000	2	118	136	157	182	208	234	260
	4	128	147	172	205	241	275	309
	6	139	161	193	239	288	338	392
	8	150	177	222	280	343	405	476
	10	163	201	251	320	397	480	-545
	12	179	220	278	365	-441	-514	-588
230000	2	123	142	167	197	226	254	280
	4	133	155	186	224	263	302	343
	6	145	172	211	261	313	366	421
	8	158	193	237	297	362	430	503
	10	173	211	263	336	419	-490	-556
	12	186	229	292	-374	-451	-522	-597

where: **the minus sign preceding the range increment indicates a result beyond the generated data**

and

h_s = **staging altitude**

γ_s = **staging flight path angle (relative)**

V_s = **staging velocity (relative)**

interpolate between values in the table to select the proper value as the staging points varies during the basic synthesis iteration process described in Section 2.3.1. If the upper bounds of the table have been exceeded during any iteration, this option provides for a default value of range to be used (via data extrapolation) so that the synthesis iteration may continue to convergence. The converged case may or may not have exceeded these bounds and thus this procedure is helpful in preventing aborted cases or at least in diagnosing problem areas. The following are the upper bounds on the staging conditions used:

h_s = 250,000 ft (or resulting apogee h = 300,000 ft)

V_s = 13,000 ft/sec

γ_s = 14 deg

The total reference range requirement in nautical miles, R_{REF}, is computed as:

$$R_{REF} = \Delta R_{LO} + \Delta R_{DATA} + DRNG$$

where,

ΔR_{LO} = range increment from the launch site to the staging point (from computed range angle - GTSM)

ΔR_{DATA} = range increment from the staging point to initiation of the flyback cruise from Table 2-9

DRNG = input specified additive range factor (n.mi) to bias results.

The parameter R_{REF} is the basic range requirement which is utilized to calculate the cruise performance mass ratio described in the next section (Section 2.3.4.2). ΔR_{LO} is supplied from the ascent trajectory simulation and is also internally stored with the staging conditions for use in this option. DRNG is usually input to account for the additional range increment required to perform the entry maneuvers over a finite period of time ($\approx$ 25 n.mi) rather than instantaneous bank and pitch attitudes assumed for the data base or to reflect the results of a detailed analysis of the entry trajectory for a particular baseline configuration.

This procedure also provides for use of an empirically derived correlation factor and which is applied to the reference range requirement, R_{REF}, to account for variations about the nominal $C_{L_{max}}$ utilized to generate the parametric range increments, ΔR_{DATA}:

$$R'_{REF} = R_{REF*} \left[\frac{0.723}{CLVG} \right]^{0.25}$$

where, R'_{REF} = adjusted value of range requirement, R_{REF}

CLVG = input specified adjustment factor if $C_{L_{max}}$ is not ≈ 0.7

This adjustment is made only if input value of CLVG $\neq 1$.

This procedure for defining the cruise range requirement for the booster is obtained by setting the input flag FLYBCK=1. This option has been successfully used for a wide variety of shuttle configurations with a resultant cruise range of within ± 10% of a detailed analysis.

b. Staging Q Function Range. This option provides for determining the cruise range requirement for flyback to the launch site with the use of an empirically derived formula based on the dynamic pressure at staging, Q_s. The form of the equation was derived from predesign observations of the effect of Q_s on typical booster entry profiles and the resulting flyback range requirements. Use of this option requires that the initial pitchover maneuver performed during flight phase 1 of the nominal ascent trajectory (Section 2.3.1.2) be used to target to an input specified staging dynamic pressure (terminal state of flight phase 3), however.

The reference range requirement in nautical miles, R_{REF}, is computed as:

$$R_{REF} = -55.915 * \log_{10} (Q_s) + 430 + DRNG$$

where, DRNG = input specified additive range factor (n. mi) to bias results from basic form of equation.

This procedure is obtained by setting the input flag FLYBCK=2. This option is particularly useful in predesign where minimal information is available on the hypersonic entry characteristics of the booster configuration.

c. Constant Range. This option provides for specifying the cruise range requirement with the constant input. The reference range requirement, R_{REF}, is specified during each basic iteration as:

$$R_{REF} = DRNG$$

where, DRNG is input specified (n. mi).

This option is obtained by setting the input flag FLYBCK=3. This option is helpful in predesign where use of inflight refueling with minimum cruise range or preselected distances to downrange landing sites are, a priori, known.

d. Ballistic Impact Range. This option provides for determining the cruise range requirement as a function of the intersection of the conic orbit which is defined by the conditions at staging with the surface of the earth. The reference range requirement in nautical miles, R_{REF}, is specified during each basic iteration as:

$$R_{REF} = \Delta R_{IMP} + \Delta R_{LO} + DRNG$$

where,

ΔR_{IMP} = range increment from staging point to predicted impact point on earth surface (from computed IIP central angle - GTSM, Section 2.2.2)

ΔR_{LO} = range increment from the launch site to the staging point (from computed range angle - GTSM)

DRNG = input specified additive range factor (n.mi) to bias results.

This option is obtained by setting the input flag FLYBCK=4. This option is useful in predesign to estimate the range requirement to downrange landing sites or as a "first-cut" estimate at the requirement for full flyback to the launch site.

e. Entry Trajectory Simulation Range. The booster entry/return trajectory can be computed by numerical integration of the flight differential equations of motion by specifying the input FLYBCK=5. This option uses the same trajectory simulation module (GTSM) that is used for the simulation of the ascent trajectories, however in this case the trajectory profile, vehicle control, and vehicle modeling are NOT fixed (with the exception of the aerodynamic reference areas which are internally set equal to the computed booster theoretical wing area), and thus must be input by the user. It is emphasized that care must be taken when defining the simulated return trajectory since errors can cause termination of the case. The most frequent error is the specification of a flight section termination condition which is not satisfied soon enough to prevent impact with the earth. Avoidance of this type of error requires a priori knowledge of staging conditions and the return trajectory. Consequently it is recommended that the numerically integrated return be used only after a similar case which utilizes another return option has been generated.*

*Also cases in which the numerically integrated return option is specified will generally require from 10 to 40 per cent more computer run time.

The booster return flight is assumed to be initiated from the booster state and attitude at the staging point. These entry "initial" conditions are defined at the end of simulation section 3 of the ascent trajectory. The state and attitude parameters which are defined at this point are presented in the following list; the quantities in parentheses are the associated program internal parameters.

t	Absolute time	(T, SW(20))
R	Geocentric radius magnitude	(Z(1), SW(37))
V	Relative velocity magnitude	(Z(2), SV(8))
γ	Relative flight path angle	(Z(3), SV(10))
β	Relative azimuth	(Z(4), SV(31))
θ	Geocentric latitude	(Z(5), SV(32))
W	Booster weight at entry/cruise conditions	(Z(7), SV(34))
h	Altitude above the oblate earth	(Z(8), SV(9))
σ	Bank angle	(Z(11), SV(38))
α	Pitch angle of attack	(Z(12), SV(39))
λ	Yaw angle of attack	(Z(13), SV(40))
ψ	Pitch angle as measured from the geocentric radius vector	(Z(43), SV(41))
ϕ	Geocentric longitude	(Z(6), SV(33))

The numerically integrated booster return trajectory is initiated in simulation section 8 (Sections 1 through 7 are allotted to the ascent trajectory). Up to 8 simulation sections (sections 8 through 15) and the last two general iteration blocks (G.I.S. blocks 3 and 4) are available for the return trajectory. Specifically this means that for the return trajectory, the values of subscripts K and M which are described in Section 4.2.3 $DATA1 are restricted such that

$K = 8, 9, 10, 11, 12, 13, 14,$ or 15

and $M = 3$ or 4

None of the GIS blocks or sections 9 or above need be used when the integrated return trajectory option is used. The number of simulation sections to be computed is input into $DATA1 via the SEC parameter. The value of SEC is the total number of simulation sections (ascent plus return sections) to be computed.

If SEC < 7., SEC is internally set to 7 and only the ascent trajectory is computed.

If SEC = n where 7 < n ≤ 15, the return trajectory is simulated with n - SEC simulation sections(assumed FLYBCK = 5.,)

Up to two tables for both the aerodynamic axial coefficient and the aerodynamic normal coefficient are available (see Section 4.2.3.6 , Aerodynamic Models). These tables are tables 4 or 5 which require that the table subscript L (see sec. 4.2.3 $DATA1) be equal to either 4 or 5. It is possible to use tables 1, 2 or 3, but in this case the user should note that tables 1, 2, or 3 are tables defined for ascent configurations* consequently selection of one of these tables for the return trajectory is equivalent to specifying that the return model in question be equivalent to that used for one of the ascent configurations. The corresponding booster return aerodynamic reference areas are internally set equal to the value of the computed theoretical wing area of the booster. These parameters together with the associated program internal parameters (the latter appears in parentheses) are:

A_{ξ}	Aerodynamic axial reference area	(Q(5, K), TB45)
A_{ζ}	Aerodynamic normal reference area	(Q(6, K), TB45)

for K = 8, 9, ..., 15

Three attitude tables which model either pitch angle as a function of time or pitch angle of attack as a function of Mach number are available during the

*The ascent configurations are (1) powered booster in the presence of the attached orbiter (Table 1), (2) no thrust orbiter in the presence of the attached booster (Table 2), and (3) powered solo orbiter (Table 3).

numerically integrated booster return flight. These tables are tables 2,3, or 4, consequently the table subscript L (see Section 4.2.3, \$DATA1) must be equal to either 2, 3, or 4.

Two propulsion thrust-time tables are likewise available to the booster return flight simulation. These tables are tables 1 and 2 with associated admissible values of L being 1 or 2.

Upon completion of the numerically integrated booster entry/return trajectory, the range from the specified target or landing site (RNG_T | end of integrated flight) is defined (see Sections 2.2.2.10, Position in Radar Coordinates, and 4.2.3.3, Targeting for a discussion of the range calculation and the definition of the target/landing site). The reference booster range requirement (R_{REF}) which is used in the booster cruise performance calculation (discussed in Section 2.3.4.2) is then defined

$$R_{REF} = DRNG + RNG_T \,\big|\, \text{end of integrated flight}$$

where

DRNG is a \$DATA2 input additive range parameter (SQ(10, 3))

RNG_T is the computed range to the landing/target site from the vehicle (Z(95))

Two typical examples of integrated return trajectories are presented in the following paragraphs. The first example assumes that the bank angle (σ) and the pitch angle of attack (α) can be modeled by linear segments where each segment is used to define a simulation section. The second example utilizes trajectory segments in which the time rate of change of relative flight path angle is held constant by modulation of the bank angle. In this case, the booster pitch angle-of-attack is assumed to be a function of Mach number. In both examples, the vehicle is headed toward a specified landing site prior to the final transition and cruise.

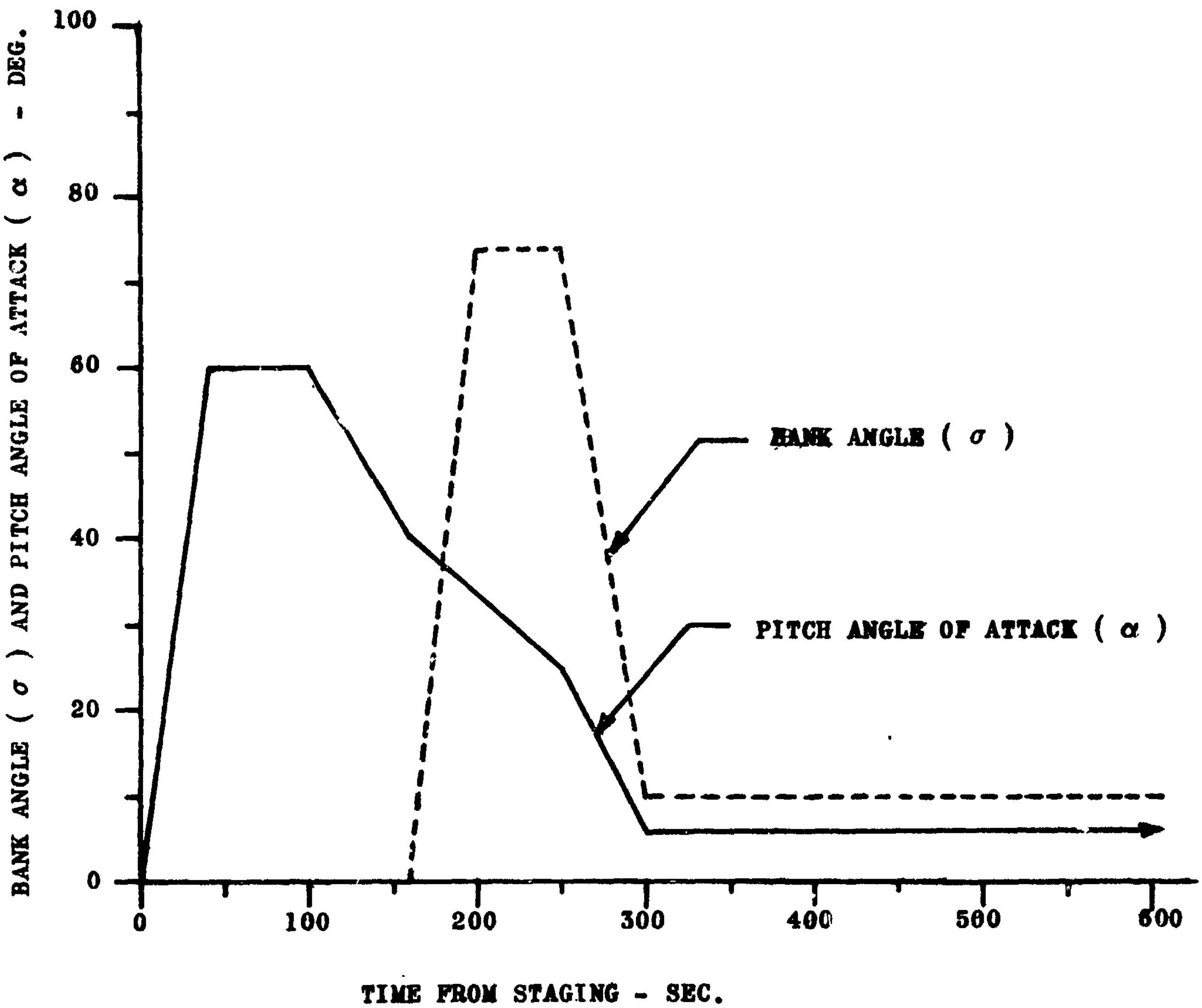


FIG. 2-21 LINEAR ATTITUDE ANGLES FOR A BOOSTER ENTRY/RETURN TRAJECTORY

Example 1. The bank angle (σ) and the pitch angle-of-attack (α) can be approximately modeled by the linear segments shown in Fig. 2-21. The return simulation is to be initiated at booster staging which from previous SSSP cases is known to occur at approximate values of 220,000 ft altitude (h), 10,000 ft/sec relative velocity (V), and 6 deg relative flight path angle. The mission requires a right turn until the booster is headed toward a landing site at 35.33°N, 232.03°E. The numerically integrated return trajectory is terminated when the vehicle descends to a 40,000 ft altitude.

With this information, a flight profile is constructed. Inspection of Figure 2-21 reveals that eight simulation sections are required to provide proper altitude modeling. Fortunately there are no other modeling discontinuities requiring the use of other sections as there are only 8 simulation sections available for the return simulation. The total number of simulation sections is thus 15, (7 for the ascent plus 8 for the return), consequently the impact parameter SEC is input as "SEC=15.," Since the attitude angles σ and α are linear functions, the $DATA1 input parameters SIGC(K) and SLPC(K) can have values of 1. or 2. for SIGC(K) or 1., 2., 4., or 5. for ALPC(K) (See Tables 4-3 and 4-6 in Section 4.2.3.6 Attitude Models). The selection of which of these values to use is optional, however the user should be aware of the differences between these codes in order to insure that the input is consistent. The following disucssion assumes that both SIGC(K) and ALPC(K) have a value of 1. unless otherwise noted. It is noted that the subscript K denotes the number of the simulation section. The following discussion includes a few of the required input parameters in order to familiarize the user to the input procedures. Most of the input is NOT mentioned in the discussion. The flight profile is:

1. Section 8 is initiated at staging. The bank angle is zero (constant) until 160 seconds, however α increases linearly to 60 deg when the time from staging is 40 seconds, hence Section 8 must be terminated either by specifying a termination relative time (t_r) of 40 seconds

or termination α of 60 degrees (either option may be used). The required $DATA1 input (ALPC(8), ALPDT(8), and SALP(8) for the α expression (see Section 4.2.3.6, Attitude Models) are defined

$$\alpha = \text{ALPDT}(8) * t_R + \text{SALP}(8)$$

where

ALPC(8) = 1., (see footnote)

ALPDT(8) = 1.5, (60 deg/40 sec)

SALP (8) = 0., (see footnote)

The corresponding $DATA1 parameters which define σ (SIGC(8)), SIGDT(8) and SSIG(8) need not be input * since their values should be 1., 0., and 0., respectively in order to specify the constant zero bank angle.

Similarly the termination parameters STGC(8), STGD(8), and STGT(8) need not be input * since Section 8 can be terminated on t_R. In this case, "STGV(8) = 40.," should be input.

2. Section 9 which is initiated at the time at which Section 8 is terminated determines the trajectory during the time when α = 60 degrees and is consequently terminated when t_R = 60 seconds (t_R is the section relative time). For this section, "SALP(9) = 60.," and "STGV(9) = 60.," must be input.

3. Section 10 must be terminated when the third α linea segment ends; this occurs at t_R = 60 seconds, consequently STGV(10) = 60., must be input. In this case "SALP(10) = 60.," (the value of α at the initiation

*This value is the same as the internally compiled default value and consequently need not be input if it has not been specifically input previously to the current $DATA1 input block.

of Section 10) and "ALPDT(10) = -.333333, (-20 degrees/60 seconds, ncte the minus sign). The bank angle is zero throughout this section and need not be input*.

4. Section 11 includes both a $\dot{\sigma}$ and an $\dot{\alpha}$ term, consequently the following altitude parameters are input:

   SIGDT(11) = 1.85,
   ALPDT(11) = -.16667,
   SALP(11) = 40.,

   SSIG(11) need not be input since its required value (0.) is the internal default value. This section is terminated when σ = 74 degrees which occurs at t_r = 40 sec.

5. Section 12 is terminated after 50 seconds (t_r = 50.). In this case (SSIG(12) = 74., must be input unless the "SIGC(11)=2.," option is used (see Sec. 4.2.3.6 attitude Models) and "SIGDT(12)=0.,". Since the termination time occurs during a linear α segment, it is more convenient (though not necessary) to specify the "ALPC(12)=2." option which references the value of α at the termination of the previous section. If this is done, the following values are input:

   ALPC(12) = 2.,
   ALPDT(12) = -.166667,
   SALP(12) = 0.,

   Note that SALP(12) must be equal to zero in this case since the starting value of α for Section 12 is set equal to the final value of α in Section 11.

6. Section 13 is characterized by negative $\dot{\sigma}$ and $\dot{\alpha}$ rates. These are input by the procedures used for the previous sections. This section is terminated at t_r = 50 sec.

7. Section 14 is flown at a constant α of 6 degrees and a constant σ of 10 degrees. This section completes the turning maneuver ($\sigma > o$) which was initiated in Section 11, consequently Section 14 is terminated at a relative azimuth value such that the vehicle is headed toward the target/landing site which is defined by $DATA1 input parameters TLAT(M) and TLNG(M). This occurs when the difference between the azimuth to the target/landing site from the vehicle and the azimuth of the vehicle (ΔAZ_V) (see Section 2.2.2.10 Position in Radar Coordinates) is zero. Since the booster is banked to the right, its azimuth is expected to continually increase while the azimuth to the target/landing site should remain nearly the same, the value of ΔAZ_V is thus algebraically decreasing. Additionally, if the vehicle azimuth takes on values near 0 or 360 degrees, it is necessary to consider the principal cycle discontinuity which occurs at 0 and 360 degrees. There is an optional $DATA1 input flag (PRIF(K)) which can be used to allow the azimuth to take on either negative values or values greater than 360 degrees (see Section 4.2.3.5 Program Control). Based on this discussion, the following $DATA1 parameters are input with values:

STGC(14) = 94.,
STGD(14) = -1.,
STGT(14) = .001, (need not be input)*
STGV(14) = 0., (need not be input)*
PRIF(14) = 1.,

* This value is the same as the internally compiled default value and consequently need not be input if it has not been specifically input previously to the current $DATA1 input block.

8. Section 15 is the final section. It is terminated when the booster descends to an altitude of 40000 ft. The booster is flown at a constant α of 6 degrees (see Fig. 2-21), but the bank angle is zero since the vehicle is already headed toward the target/landing site. The required section termination parameters (as input to $DATA1) are:

   STGC(15) = 8.,
   STGD(15) = -1.,
   STGT(15) = 1.,
   STGV(15) = 40000.,

   Example 2 In this example, two convenient but physically meaningful techniques are employed to define the α and σ time histories. Use of these options generally results in a smoother trajectory (eliminates any oscillatory phugoid motion) and a reduction in the number of required simulation sections. The first technique is to assume that the desired or required value of α is dependent* on the Mach number (see Section 4.2.3.6 Program Control for input instructions). The second technique is to assume that during the banked turning segment of the flight, the bank angle (σ) is modulated in order to hold a specified constant time rate of change of relative flight path angle $\dot{\gamma}$. Additionally this program has the capability to hold a constant $\dot{\gamma}$ with a fixed σ by modulation of α, however, this latter option is not used in this particular example. Both the constant $\dot{\gamma}$ options are described in Sections 2.2.2.6 Specified Time Rate of Change of Relative Flight Path Angle and 4.2.3.6 Program Control. As in the case of example 1, only a few of the required input parameters are discussed, however a complete listing of the required booster entry/return $DATA1 input parameters is shown in Table 2-10. The flight

* The trim angle of attack (α_T) is conveniently expressed as a function of Mach number, consequently if the vehicle is constrained to fly at α_T for any time, this option should be used.

Table 2-10. Sample Booster Numerically Integrated Return Trajectory Input

```
SEC       =        13.,
CRGF(3)=           4.,
TLAT(3)=          35.3333,
TLNG(3)=         232.034,
HTL8(8)= 6*100000.,
STEP(8)=        6*20.,
TOL8(8)= 6*500000.,
PRIF(8)=         6*1.,
STGC(8)=           0.,          0.,          3.,         8.,         94.,          8.,
STGD(8)=           1.,          1.,         -1.,        -1.,         -1.,         -1.,
STGT(8)=           0.001,       0.001,       0.0005,     1.,          0.001,       1.,
STGV(8)=          10.00,       30.00,       -5.00, 100000.,           0.00, 40000.,
WOUT(8)=         6*3.,
XOUT(8)=         6*1.,
ATMC(8)=         6*1.,
AC2(8)=          6*4.,
AC3(8)=          6*4.,
A2(1,4)=        -200.,        200.,
C2(1,1,4)=         -.0231678,-.0231678,-.0231678,-.2000000,-.1385019,-.1644357,
C2(7,1,4)=         -.1789872,-.1818783,-.1308036,-.1308036,
C2(1,2,4)=         -.0231678,-.0231678,-.0231678,-.2000000,-.1385019,-.1644357,
C2(7,2,4)=         -.1789872,-.1818783,-.1308036,-.1308036,
XKA2(1,4)=        10.,          2.,
XM2(1,4)=         -1.0,         0.0,         0.8,        1.0,         2.0,         4.0,
XM2(7,4)=          6.0,         8.0,        10.0,      100.0,
A3(1,4)=        -200.,        200.,
C3(1,1,4)=         -.5755743,-.5755743,-.5755743,-.2000000,-.5508332,-.9551889,
C3(7,1,4)=        -1.340583,-1.721960,-1.973959,-1.973959,
C3(1,2,4)=         -.5755743,-.5755743,-.5755743,-.2000000,-.5508332,-.9551889,
C3(7,2,4)=        -1.340583,-1.721960,-1.973959,-1.973959,
XKA3(1,4)=        10.,          2.,
XM3(1,4)=         -1.0,         0.0,         0.8,        1.0,         2.0,         4.0,
XM3(7,4)=          6.0,         8.0,        10.0,      100.0,
GDOT(8)=           0.,          0.,         -0.100,      0.,          0.,          0.,
SIGC(8)=           1.,          1.,          3.,         3.,          3.,          1.,
SIGDT(8)=          0.,          0.,          1.,         1.,          1.,          0.,
SSIG(8)=           0.,          0.,          0.,         0.,         35.,          0.,
ALPC(8)=           1.,          1.,        4*26.,
ALPDT(8)=          2.00,        2.00,      4*0.00,
PDT(1,3)=          6.00,        6.00,        6.00,       0.00,       12.00,       30.00,
PDT(7,3)=         44.00,       54.00,       60.00,      60.00,
PT(1,3)=          -1.00,        0.00,        0.80,       1.00,        2.00,        4.00,
PT(7,3)=           6.00,        8.00,       10.00,     100.00,
SALP(8)=         -20.00,        0.00,      4*0.00,
XKPT(3)=          10.,
XOUT(8)=          10.,        3*2.,        2*5.,
```

profile is:

1. Section 8 is initiated at staging. The bank angle is zero (constant) and the booster is pitched-up at the rate of 2 deg/sec. These models are the linear segmented altitude angle options which were used in example 1 (SIGC(8) and ALPC(8) equal 1.,). This section is terminated after 10 seconds.

2. Section 9 is a continuation of the flight described by Section 8 (this section was inserted into the flight to provide an available section if an additional maneuver was to be modeled in a subsequent case). This section is terminated when α equals 60 degrees (equivalent to t_r = 30 sec.).

3. Section 10 utilizes both the α as a function of Mach number option and the constant $\dot{\gamma}$ by σ modulation option. This section is intended to simulate the flight through its apogee and terminate upon entry at a relative flight path angle (γ) of -5 deg. The $\dot{\gamma}$ value was selected to be -0.10 deg/sec; this parameter could be optimized if required. The booster is flown at the trim angle of attack (α_T) which is a function of Mach number (note the input of the α-Mach number table, \$DATA1 parameters PDT(I, L), PT(I, L), XKPT(L), and "ALPC(10) = 26.," where L denotes the table number and I denotes the table element number). Other \$DATA1 input parameters which are noted include

GDOT(10)	=	-0.100,	
SIGC(10)	=	3.,	
SIGDT(10)	=	1.,	(denotes right bank)
SSIG(10)	=	0.,	(denotes minimum acceptable value of σ)
STGC(10)	=	3.,	
STGD(10)	=	-1.,	
STGT(10)	=	0.0005,	
STGV(10)	=	-5.00,	

4. Section 11 is similar to section 10 except that $\dot{\gamma}$ is maintained at 0 deg/sec. This section is terminated upon descent to a 100000 ft altitude.

5. Section 12 continues the flight described by Section 11 and as in the case of Section 9, Section 11 is an extra section which is provided in order to be available for any additional maneuvers to be modeled in subsequent cases. This section is terminated when the booster is headed toward the landing/target site defined by $DATA1 input parameters TLAT(3) and TLNG(3) (see example 1 simulation section 14 for more details). The minimum allowable bank angle is 35, consequently "SSIG(12) = 35.," is input.

6. Section 13 is the final trajectory segment in which the booster flys toward the target/landing site while descending to 40000 ft. The booster is consequently flown with a zero bank angle and the section is terminated at a descending altitude of 40000 ft.

From the previous examples, it is noted that some prior knowledge of the return trajectory profile is necessary in order to define meaningful simulation sections which will shape the trajectory to required position without impacting the earth. The use of the constant $\dot{\gamma}$ and α-Mach number options simplifies the determination of simulation control flags and values (e.g. section termination conditions).

2.3.4.2 Booster Cruise Performance Calculation. The required cruise performance mass ratio for the booster return flight is computed by using a sequence of simplified closed form experessions. There are three options for these calculations. For each of these options, it is necessary to have previously defined the required booster range capability (R_{REF}) from a reference point in the return trajectory (see Section 2.3.4.1). These options are described in the following paragraphs.

a. Simplified Single Segment Cruise. This option assumes that the booster is operated in the cruise mode for a range equal to R_{REF}. The Brequet range equation is used to determine the cruise performance mass ratio (μ_c) and the cruise performance parameter (ν_c).

$$\mu_c = \left(\frac{W_I}{W_I - WFLYX} \right) \quad \exp \left[\frac{1.689 * R_{REF} * SFC}{(VCRUSE * ALD)} \right]$$

$$\nu_c = \mu_c - 1$$

where

W_I	= weight of booster just prior to ignition of the cruise engines from the previous synthesis iteration (lb)
WFLYX	= input specified optional weight additive term (lb)
R_{REF}	= required booster range capability from a reference point in the return trajectory (n. mi)
SFC	= input specified booster specific fuel consumption in the cruise mode (lb/lb-hr)
VCRUSE	= input specified cruise velocity (fps)
ALD	= input specified booster lift-to drag ratio during cruise
1.689	= the conversion constant from knots to ft/sec.

This procedure is obtained by setting the input flag KBFUEL = 1.

b. Four Segment Reference Range Option 1. Both this option and the succeeding option assume that the required booster range (R_{REF}) from a reference point in the return trajectory is comprised of four segments. These segments are:

1. Transition
2. Idle Descent
3. Cruiseback
4. Final Descent

The transition segment is a descending glide with no fuel expenditure (booster weight is constant). The range (RT) traversed during transition can be input specified to be included in the determination of the cruiseback range. The fuel expended during idle descent (ΔW_{f_1}) is determined by assuming a constant fuel weight - vehicle weight partial derivative.

$$(\Delta W_{f_1}) = W_T * CA$$

where:

W_T = booster weight during transition (immediately prior to the idle descent

$$CA = \left.\frac{\partial W_f}{\partial W_I}\right|_{\text{idle descent}} \text{, input specified}$$

W_f = required fuel weight during a flight segment

W_I = booster weight at the initiation of a flight segment

The booster weight (W_2) upon completion of the idle descent (at the initiation of the cruiseback segment) is

$$W_2 = W_T - (\Delta W_f)_1$$

The required cruiseback range (R_2) is the reference range (R_{REF}) decremented by the ranges traversed during transition, idle descent, and the final descent.

$$R_2 = R_{REF} - RT - R1 - R3$$

where R_3 is the input specified range traversed during the final descent

R_1 is the input specified range transversed during the idle descent

The fuel expended during the cruiseback (ΔW_{f_2}) is determined from the Breguet range equation

$$(\Delta W_f)_2 = W_2 * \left\{ 1 - \frac{1}{\exp\left[\frac{1.689 * R_2 * SFC2}{ALD2 * VFLY2}\right]} \right\}$$

where

SFC2 = input specified booster specific fuel consumption during cruiseback (lb/lb-hr)

ALD2 = input specified booster lift-to-drag ratio during cruiseback

VFLY2 = input specified booster velocity during cruiseback (fps)

The booster weight (W_3) upon completion of the cruiseback (at the initiation of the final descent) is

$$W_3 = W_2 - (\Delta W_f)_2$$

Both expended weight and traversed range during final descent can also be included in the determination of μ_c and ν_c. The traversed range (R3) can be included in the determination of R_2 by the input of a non-zero value. The fuel expended during final descent $(\Delta W_f)_3$ is determined by assuming a constant fuel weight - vehicle weight partial derivative

$$(\Delta W_f)_3 = W_3 * CB$$

where

$$CB = \left.\frac{\partial W_f}{\partial W_I}\right|_{\text{final descent}}, \text{ input specified}$$

The landing weight (W_L) and the total fuel expended during the traverse of R_{REF} are

$$W_L = W_3 - (\Delta W_f)_3 - WFLYX_3$$

$$(\Delta W_f)_{Total} = WFLYX + \sum_{k=1} (\Delta W_f)_K$$

where

WFLYX = input specified optional weight additive term.

The cruise performance parameter (ν_c) and the cruise performance mass ratio (μ_c) can now be defined

$$\nu_c = \frac{(\Delta W_f)_{Total}}{W_L}$$

$$\mu_c = \nu_c + 1$$

This procedure is obtained by setting the input flag FBFUEL = 2.

c. Four Segment Reference Range Option 2. This option is the same as that described in the previous paragraph, Four Segment Reference Range Option 1, except for the equation used to define $(\Delta W_f)_1$ and $(\Delta W_f)_3$. For this option, the Breguet range equation is used to evaluate both of these parameters. Specifically,

$$(\Delta W_f)_1 = W_T * \left\{ 1 - \frac{1}{\exp\left[\frac{1.689 * R1 * SFC1}{ALD1 * VFLY1}\right]} \right\}$$

and

$$(\Delta W_f)_3 = W_3 * \left\{ 1 - \frac{1}{\exp\left[\frac{1.689 * R3 * SFC3}{ALD3 * VFLY3}\right]} \right\}$$

where the following are input specified:

SFC1 is the booster specific fuel consumption during idle descent (lb/lb-hr)

SFC3 is the booster specific fuel consumption during final descent (lb/lb-hr)

ALD1 is the booster lift-to-drag ratio during idle descent

ALD3 is the booster lift-to-drag ratio duirng final descent

VFLY1 is the booster velocity during idle descent (fps)

VFLY3 is the booster velocity during final descent (fps)

This procedure is obtained by setting the input flag FBFUEL = 3.

Special Note: The three procedures described above for determining the required cruise performance mass ratio (μ_c) and the cruise parameter (ν_c) used in the booster sizing process have special significance when one of the following Basic Synthesis Options (Section 2.3.2) are utilized:

2.3.2.4 Fixed Orbiter Gross Weight

2.3.2.5 Fixed Orbiter Propellant Weight

Both of these options adjust the booster main impulse mass ratio, μ_B, during the iteration process within the WTVOL subprogram. During these iterations an adjustment on ν_c is also performed to anticipate its effect on the overall or basic synthesis iteration process (Section 2.3.1). This adjustment not only improves convergence of the WTVOL iterations, but also of the basic synthesis iterations. The adjustment is made as follows during each Newton-Raphson iteration in WTVOL:

$$\nu_{c_{(n+1)}} = \nu_{c_{(n)}} + \left[\frac{\partial R_{REF}}{\partial \mu_B}\right] * \left[\frac{\mu_{B_{(n+1)}} - \mu_{B_{(n)}}}{R_{PAR}}\right]$$

where,

$\frac{\partial R_{REF}}{\partial \mu_B}$ = effect of changing μ_B on calculation of R_{REF} during each basic synthesis iteration. This parameter is input specified as FBPAR and has a typical value $\approx$ 250 (n. mi). Note: FBPAR is an initial estimate if FBFUEL =1. This parameter is updated during each basic synthesis iteration thereafter.

R_{PAR} = weighting factor on the ν_c update and is computed as follows:
if FBFUEL = 1.,

$$R_{PAR} = \frac{ALD * VCRUSE}{1.689 * SFC}$$

if FBFUEL = 2., or = 3.,

$$R_{PAR} = \frac{ALD2 * VFLY2}{1.689 * SFC2}$$

3/ PROGRAMMING DISCUSSION

The Space Shuttle Synthesis Program (SSSP) resulted from the combination of two independent programs – GTSM (General Trajectory Simulation Module) and WTVOL (Weight/Volume Program). To reduce computer core requirements and to improve turn-around time, SSSP is structured as an overlay program with a resident overlay and three primary overlays. SSSP is characterized by three input NAMELIST blocks and prestored input parameters. Large subscripted arrays are used to define principal parameters and flags needed by the trajectory and synthesis interfacing portions of the program and to handle the input coefficients and exponents required for the weight and volume calculations. The subsections of this section contain the description of the overall organization of the program and descriptions and flowcharts of the overlays and subroutines.

3.1 PROGRAM ORGANIZATION

SSSP consists of a resident overlay (MAIN), three primary overlays (SYNTH, GTSM, WTVOL), BLOCK DATA, and 40 subroutines. The inter-relationship of the subroutines is shown in Figures 3-1 for the MAIN overlay, 3-2 for the SYNTH overlay, 3-3 for the GTSM overlay, and 3-4 for the WTVOL overlay. The SSSP subroutines and their purposes are listed below.

Subroutine	Purpose
MAIN	Driver for program (overlay (0,0)). Contains data statements for initializing variables. Calculates flyback fuel required.
STORE	Stores orbiter and booster input data and performs arithmetic operations on input geometry.
CHECK	Prints run title.
BLOCK DATA	Pre-sets data variables for trajectory simulation.
SYNTH	Driver for input routines, calls synthesis iteration routine. Driver for overlay (1,0).
FRENCH	Driver routine to determine order of reading and storing orbiter and booster weight/volume input data.

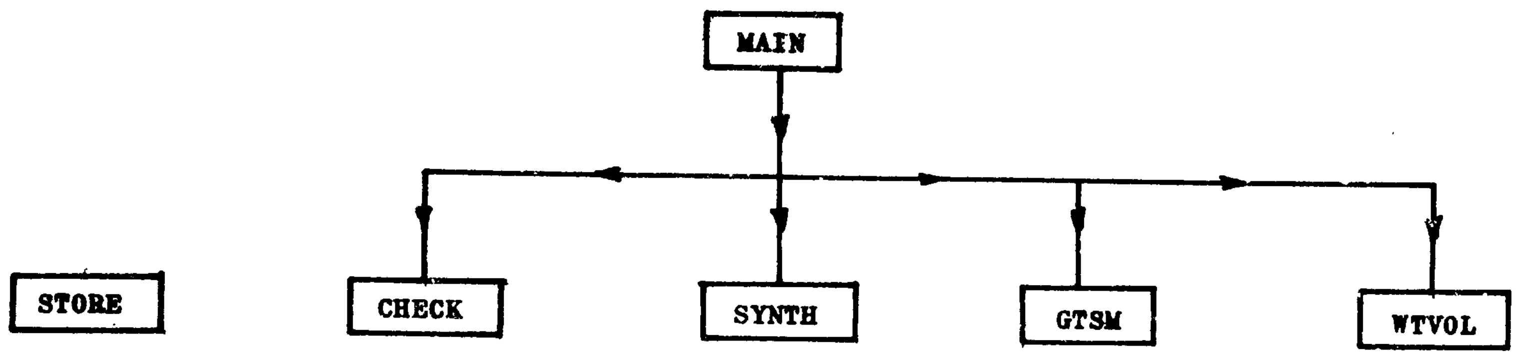


Figure 3-1 MAIN OVERLAY - SUBROUTINE INTERRELATIONSHIP

4A

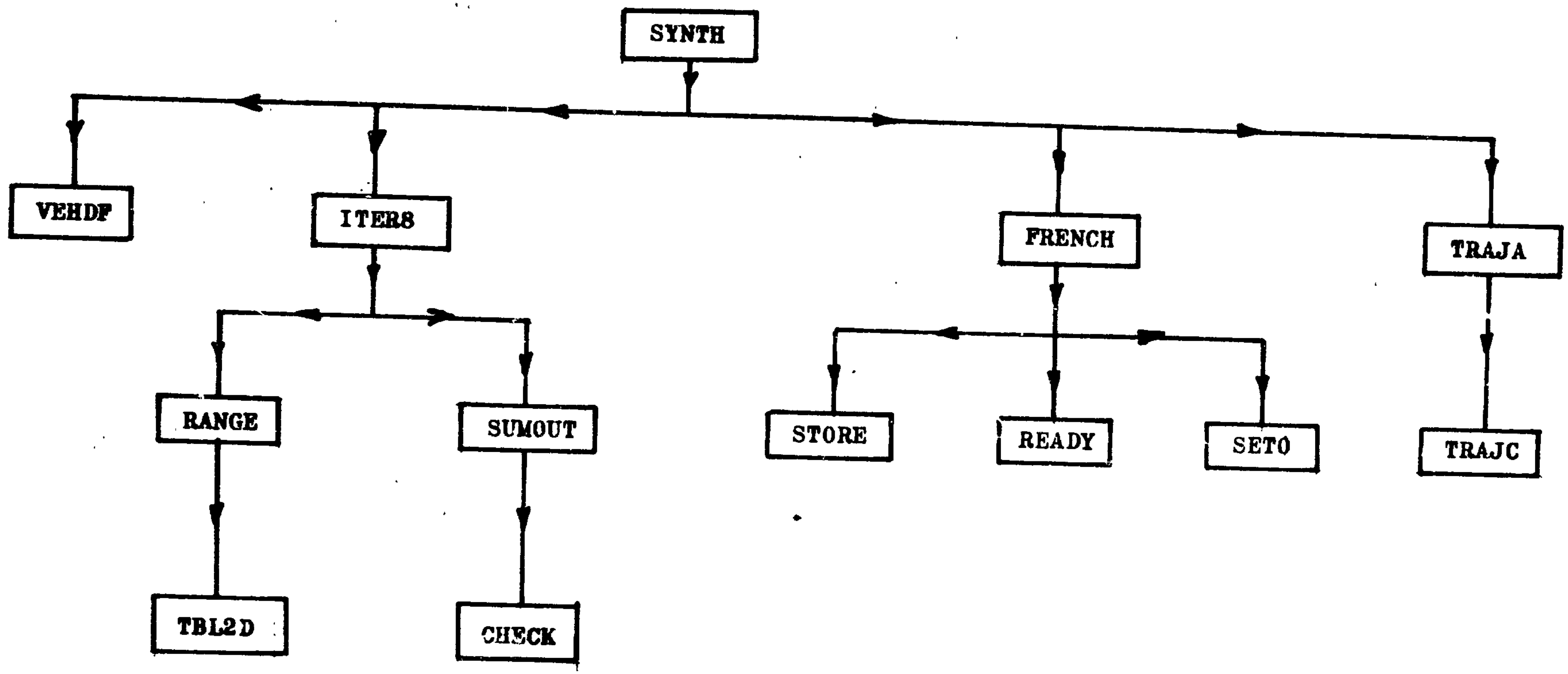


Figure 3-2 SYNTH OVERLAY - SUBROUTINE INTERRELATIONSHIP

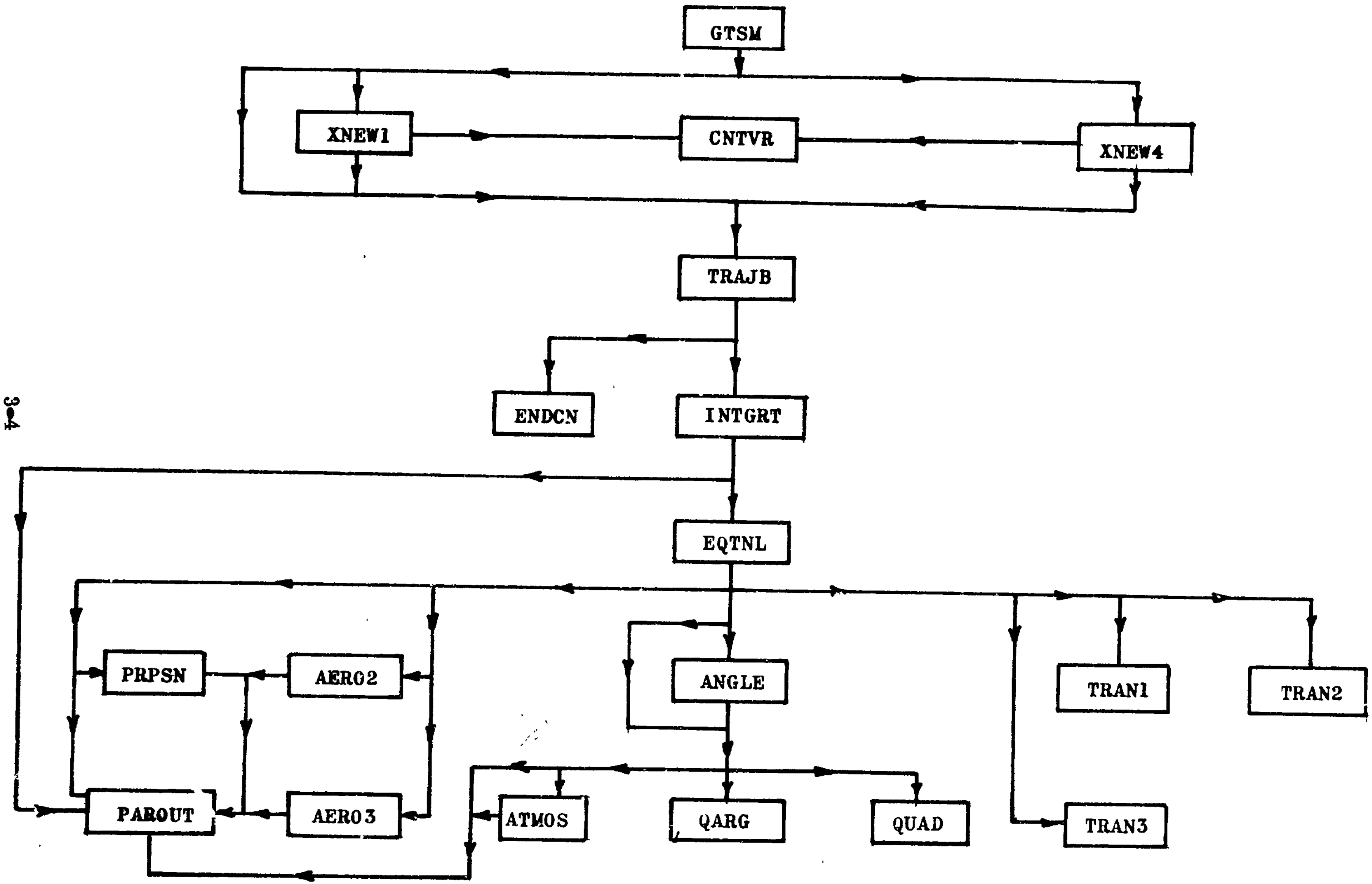


Figure 3-3 GTSM OVERLAY - SUBROUTINE INTERRELATIONSHIP

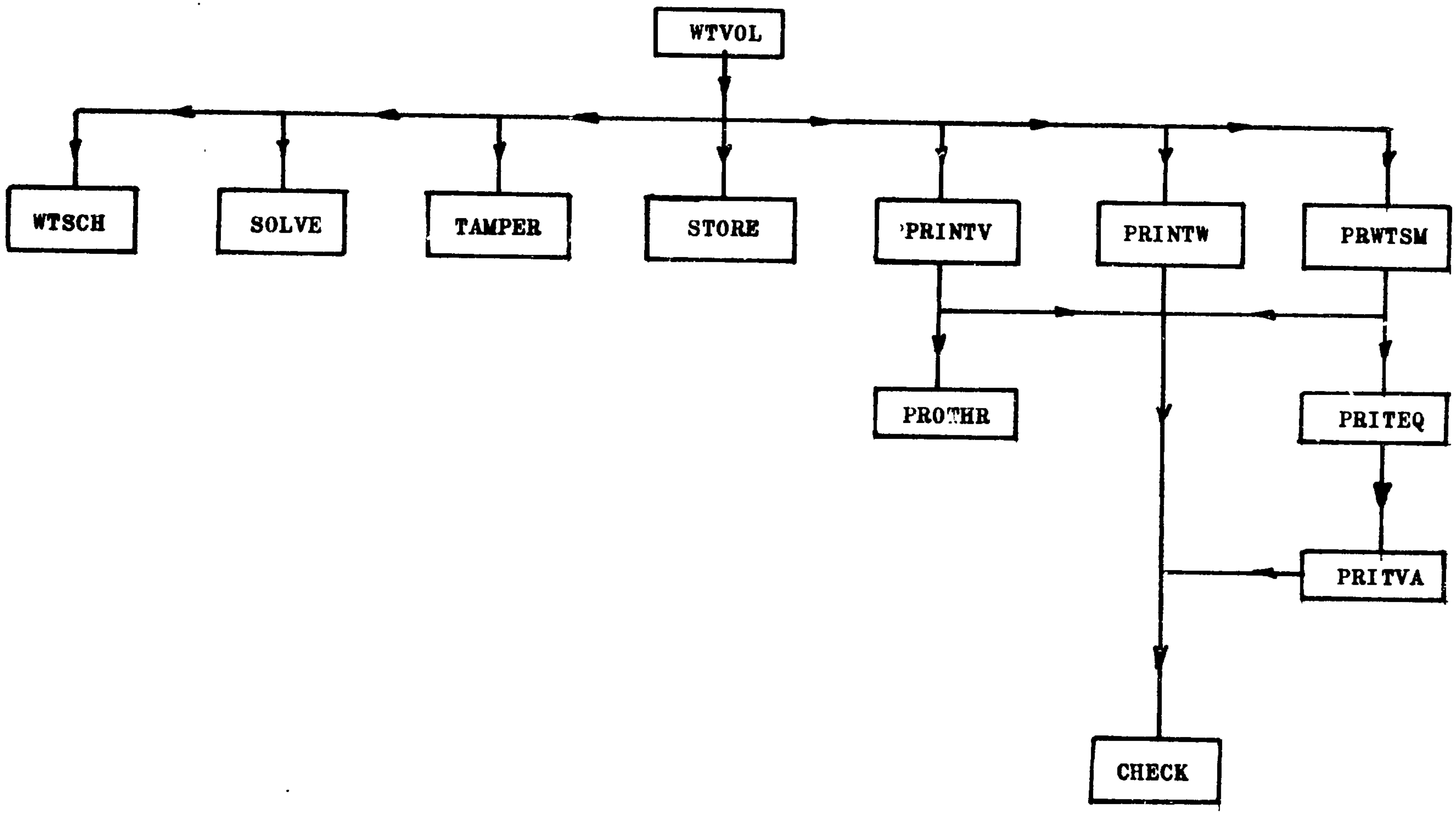


Figure 3-4 WTVOL OVERLAY - SUBROUTINE INTERRELATIONSHIP

READY	Reads input data for weight/volume routines.
SETO	Pre-sets to zero variables used for weight/volume calculations.
VEHDF	Reads "synthesis" input data and stores values in working variable arrays.
TRAJA	Storage of trajectory data, input of trajectory data, initialization of parameters, and output of input data.
TRAJC	Defines those GTSM variables which the user may not change by input.
ITER8	Checks synthesis loop for convergence, sets final print flag.
RANGE	Calculates flyback range.
TBL2D	Two dimensional table look-up routine.
SUMOUT	Prints summary sheet. Calculates thrust (nominal and uprated) for print purposes only.
GTSM	Driver for the General Iteration Scheme, defines iteration blocks and non-iteration segments of the trajectory, and runs final reference trajectory. Driver for overlay (2, 0).
CNTVR	Defines iteration control variables, stores initial values of the control variables, and re-initializes control variables.
XNEW1	(1 x 1) Newton-Raphson general iteration with steps defined by TRAJB.
XNEW4	(2 x 2) General iteration with steps defined by TRAJB.
TRAJB	Trajectory simulation driver with a sectional operating mode which causes simulation of either all the simulation sections contained in an iteration block; or all those to the start of an iteration block; or all those to the end of the trajectory. TRAJB also prints the required sectional headers and performs the bookkeeping associated with the payload and jettison weights.
ENDCN	Defines the iteration end conditions for an iteration block.

INTGRT	Kutta-Merson numerical integration scheme with general simulation section termination (general staging).
PAROUT	Output of the parameters which are computed at each integration step.
EQTNL	Provides the earth gravitational potential and surface geometry models; sub-driver which calls the atmosphere model, the three-dimensional aerodynamic models, and the propulsion models; expresses the three degree-of-freedom flight equations, the ideal velocity and associated losses acceleration equations, and the heat flux.
ANGLE	Provides the vehicle attitude model which determines the roll, pitch, and yaw attitudes as a function of time.
ATMOS	Atmosphere model which is defined by a table look-up scheme with a memory.
AERO1	Aerodynamic model which defines the yaw aerodynamic force coefficient (directed along the standard pitch axis) by means of a double table look-up scheme with a memory. This routine is currently not available in order to reduce core requirements. A dummy "do nothing" AERO1 is used in its place.
AERO2	Aerodynamic model which defines the axial aerodynamic force coefficient (directed along the standard roll axis) by means of a double table look-up scheme with a memory.
AERO3	Aerodyanmic model which defines the normal aerodynamic force coefficient (directed along the standard yaw axis) by means of a double table look-up scheme with a memory.
PRPSN	Propulsion models which define the thrust level, specific impulse, and propellant flow rates.
TRAN1	Coordinate transformation matrices M-5 and M-6.*
TRAN2	Coordinate transformation matrix M-7. *
TRAN3	Coordinate transformation matrix M-8. *

* See Section 2.2.2.2.

QUAD	Adjusts angular quantities to be within the principal cycle, from 0 to 360 degrees.
QARG	Used to call subroutine ARSIN or ARCOS or ATAN with appropriate logic to avoid program errors which occur when the subroutine argument is outside acceptable values (e.g., ARSIN (ARG) when $\|ARG\| > 1$), and to provide an angular units option (e.g., radians or degrees).
WTVOL	Driver for weight/volume routines, overlay (3,0). Contains iterations for thrust/weight, and weight options.
WTSCH	Computes vehicle weights and volumes.
SOLVE	Iteration routine for gross weight and volume in WTSCH.
TAMPER	The interface between WTVOL and GTSM overlays. Weights, thrusts, specific impulses and reference areas calculated in WTVOL are defined for GTSM variable names. Pitch rate algorithm is evaluated. Orbiter and booster weights required for summary outputs are stored.
PRINTW	Prints vehicle weight breakdown.
PROTHR	Prints design data.
PRITVA	Prints geometry coefficients, cruise back and maximum dynamic pressure conditions.
PRITEQ	Prints weight and volume coefficients.
PRWTSM	Prints weight summary.
PRINTV	Prints volumes.

3.2 OVERLAYS AND SUBROUTINES

The resident overlay, three primary overlays and 40 subroutines which comprise SSSP are described in the following subsections. Flowcharts are presented and complete FORTRAN listings appear in Appendix IV.

3.2.1 RESIDENT OVERLAY

The resident overlay, designated as OVERLAY (0,0), remains in the computer central memory throughout program execution and contains three routines, MAIN, CHECK, and STORE. The resident overlay is the communications link between the three primary overlays. Data generated by one overlay and required by another must reside in a COMMON block located on the resident overlay.

3.2.1.1 MAIN Program. The executive dirver for the SSSP MAIN, contains the logic necessary for calls to the three primary overlays, data preset by DATA statements, and equations for calculating the booster flyback fuel. A flowchart of the logic in MAIN is shown as Figure 3-5, and the FORTRAN listing appears in Appendix IV-1.

3.2.1.2 CHECK Subroutine. Printout of the user's run title is accomplished by the CHECK subroutine. Appendix IV-2 contains the FORTRAN listing.

3.2.1.3 STORE Subroutine. There are two purposes served by the STORE subroutine: (1) computation of geometry sizing coefficients which needs to be done outside the synthesis iteration loop; and (2) transfer of weight/volume data from either working variable names to discrete variable names, or vice-versa. The first purpose is accomplished by the normal entry to the subroutine, STORE. Four alternate entries (ORBSTO, BOOSTO, ORBCAL, BOOCAL) allow the transfer of data from one variable name to another.

One set of working variable names is used to read input data, calculate weights and volumes, and print the weight and volume output data, for both the orbiter and the booster. Since it is necessary to preserve the values unique to each one, two sets of unique variable names are maintained. The alternate entries ORBSTO and BOOSTO receive data under the common working variable names and store it in the unique data set. To reverse the procedure and obtain orbiter and booster data under the working variable names, the alternate entries ORBCAL and BOOCAL are used. Figure 3-6 shows a flowchart of subroutine STORE and Appendix IV-3 contains a FORTRAN listing of the subroutine.

3.2.2 SYNTHESIS OVERLAY

The synthesis overlay (overlay (1,0) is a primary overlay comprised of its driving program (SYNTH) and ten subroutines whose purpose is fourfold: (1) reading input data, (2) checking the synthesis loop for convergence, (3) calculating the booster flyback range, and (4) printing the synthesis summary data. The subparagraphs of this section describe the routines located on this overlay.

3.2.2.1 SYNTH Program. SYNTH is the short main program for the synthesis primary overlay. Its logic determines the order of the three input data categories - weight/volume, synthesis, and trajectory - by calls to the subroutines FRENCH,

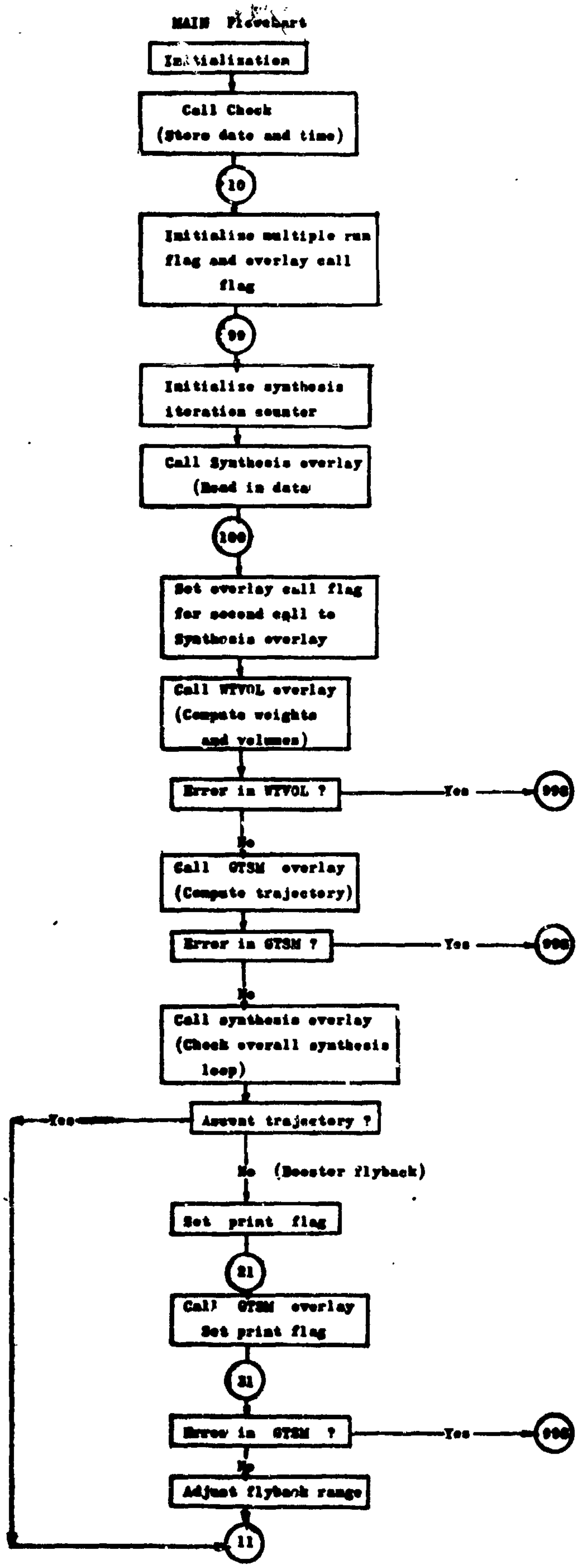


Figure 3-5 MAIN Program Flow Diagram

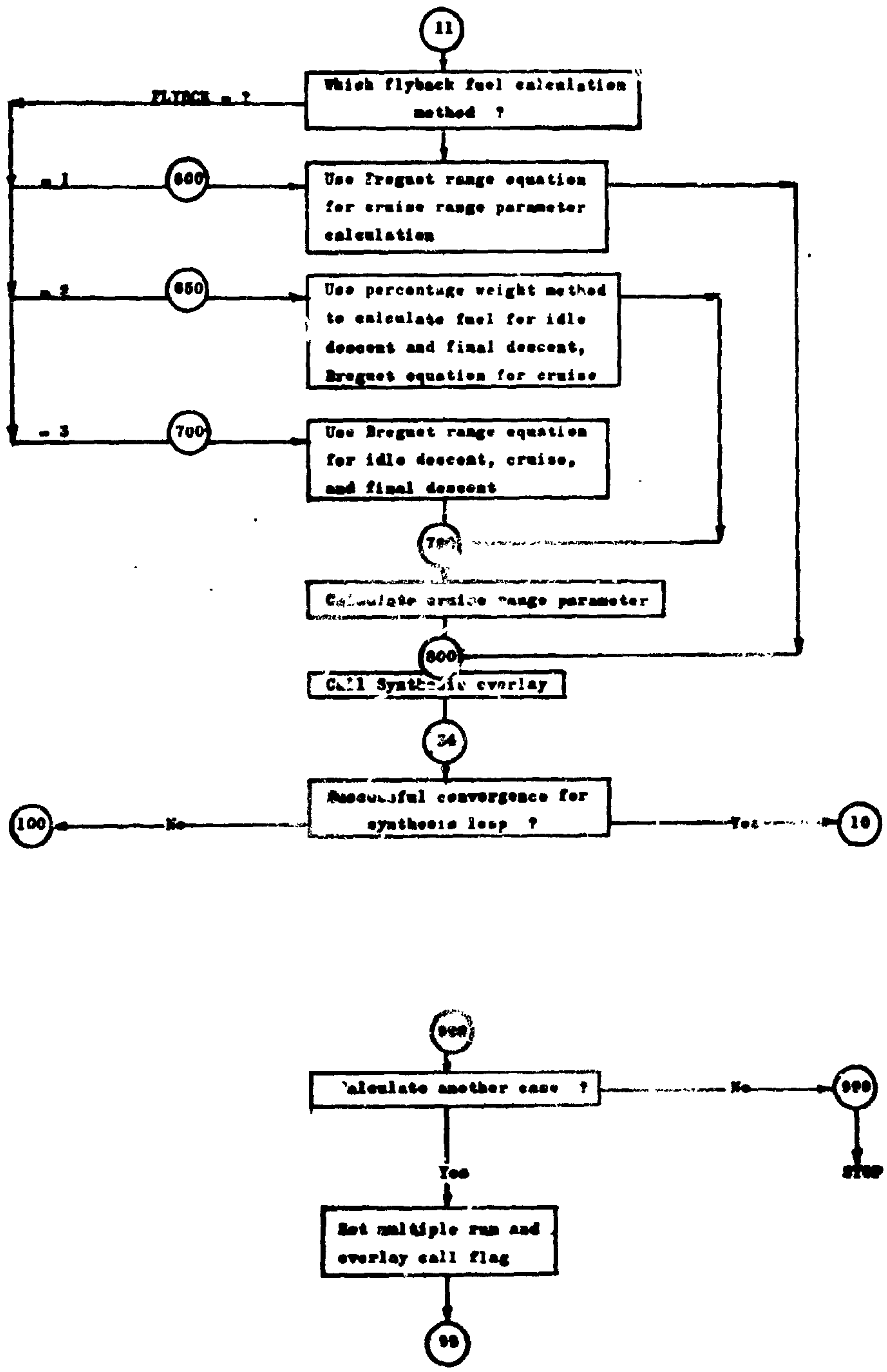


Figure 3-5 (Cont)

STORE SUBROUTINE FLOWCHART

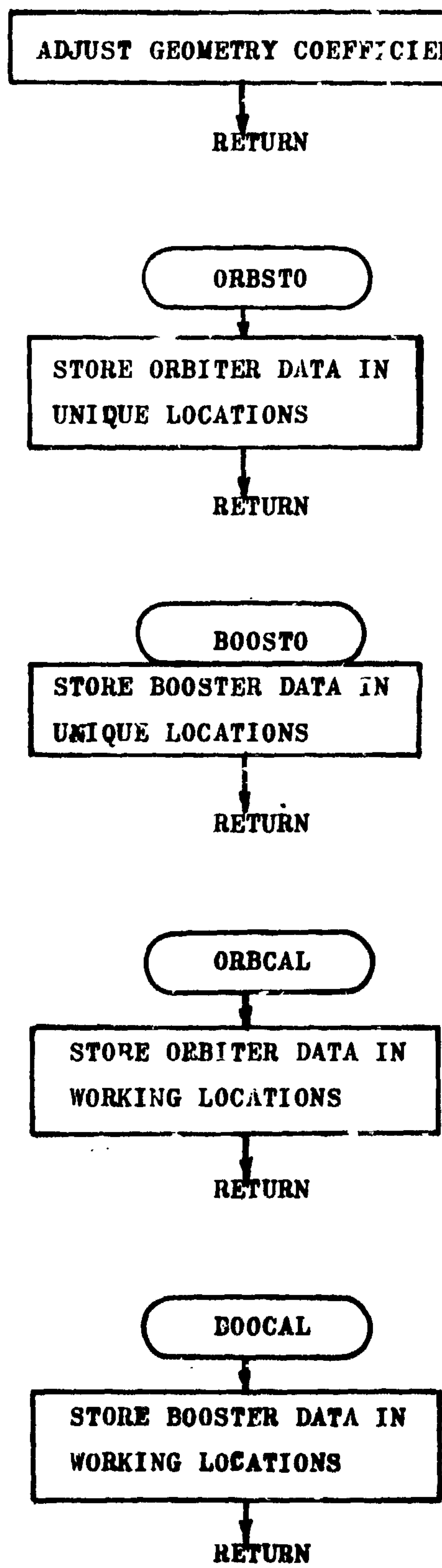


Figure 3-6 Subroutine STORE Flow Diagram

VEHDF, and TRAJA. The subroutine IRER8 is also called by this driver to check the synthesis loop convergence and when appropriate, to calculate the booster fly-back range and to print synthesis summary data. A summary flow diagram is shown in Figure 3-7 and a FORTRAN listing appears in Appendix IV-4.

3.2.2.2 FRENCH Subroutine. The subroutine FRENCH reads the "input flag" card (IOS) which carries information about the weight/volume input data. The information FRENCH thus obtains determines for the orbiter and the booster whether or not the following will occur: (1) initialization for a new vehicle (2) printing of input data, (3) punching of input data. A flow diagram is shown in Figure 3-8 and a FORTRAN listing appears in Appendix IV-5.

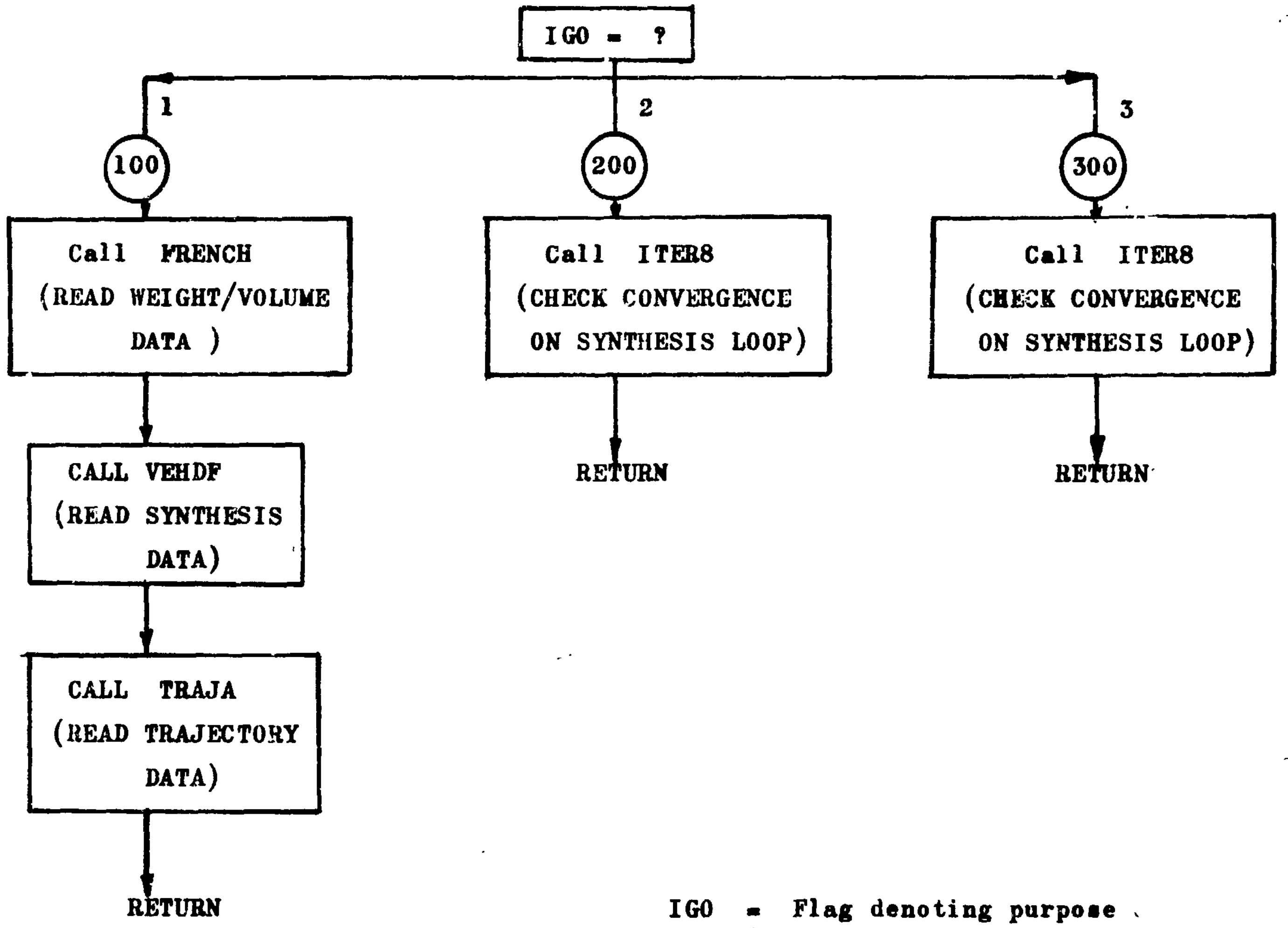
3.2.2.3 READY Subroutine. The subroutine READY handles the reading, printing, and punching of weight/volume input data for the orbiter and booster. NAMELIST statements are used for all three purposes. This routine is used in conjunction with the STORE routine and its alternate entires (sec. 3.2.1.3) to allow reading and working with orbiter and booster data by the same variable names while retaining their respective identities. A flow diagram for READY is shown in Figure 3-9 and Appendix IV-6 contains a FORTRAN listing of the subroutine.

3.2.2.4 SETO Subroutine. The subroutine SETO is used to preset to zero the variables used for weight/volume calculations. The preset value remains until changed by the input data list or subsequent calculations of a value for the variable. The FORTRAN listing for this routine appears in Appendix IV-7.

3.2.2.5 VEHDF Subroutine. The purpose of the subroutine VEHDF is to read and print the "synthesis" input data (sec. 4.2.2) and store it in working variable array names. An average specific impulse is calculated and the orbiter mass ratio is adjusted by the input mission ideal velocity for later use as an initial estimate in the weight/volume routines. If solid strap-on rockets are to be simulated, their propellant weight, sea level thrust, and total weight are calculated and stored for future reference. A flow diagram of VEHDF is shown in Figure 3-10 and a FORTRAN listing is in Appendix IV-8.

3.2.2.6 TRAJA Subroutine. The principal purpose of TRAJA is to define and initialize all data which is required for the trajectory simulation. This is accomplished by compiling-in reference input data, reading input data, initializing the parameters which are required to start the trajectory simulation, and printing all initial data. A summary flow diagram of subroutine TRAJA is shown in Figure 3-11 and the FORTRAN listing is presented in Appendix IV-9.

SYNTH SUBROUTINE FLOWCHART



IGO = Flag denoting purpose call to this overlay - to read input data or to check on convergence.

Figure 3-7 Subroutine SYNTH Flow Diagram

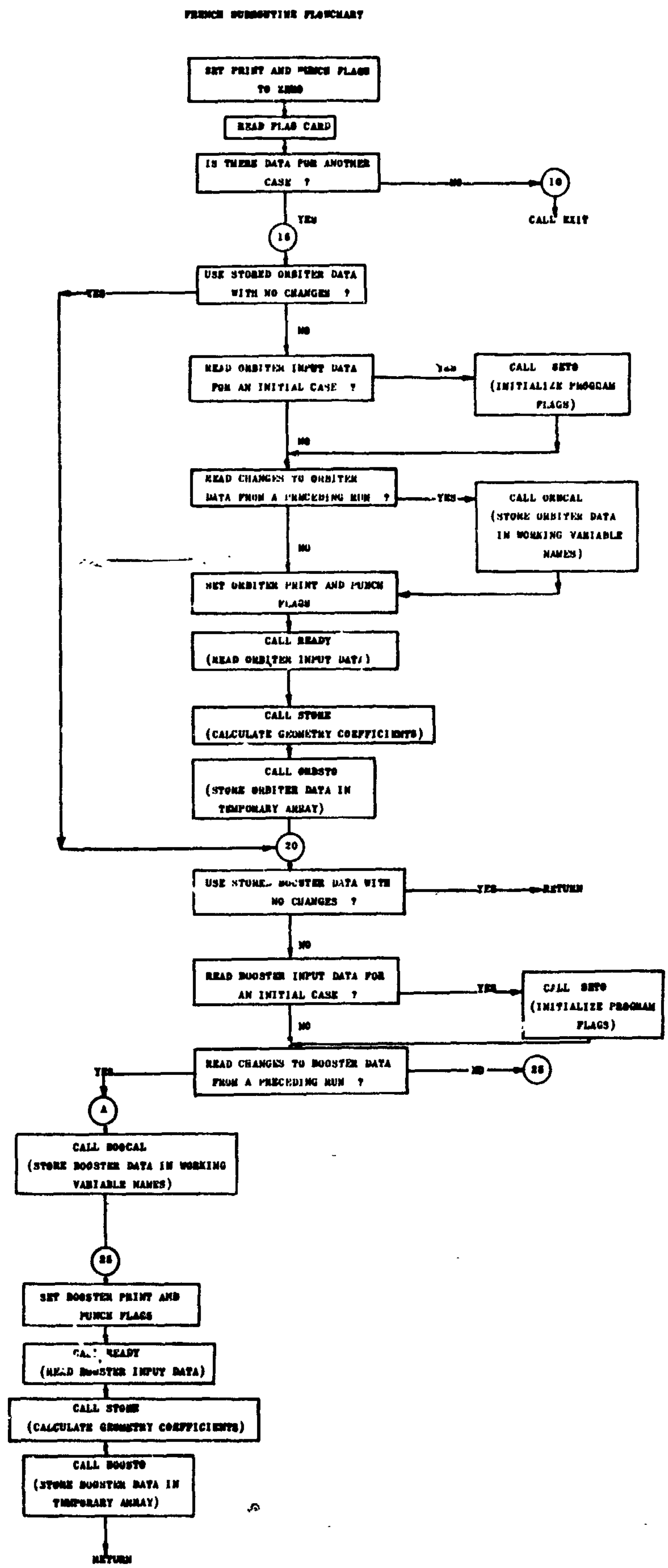


Figure 3-8 Subroutine FRENCH Flow Diagram

READY SUBROUTINE FLOWCHART

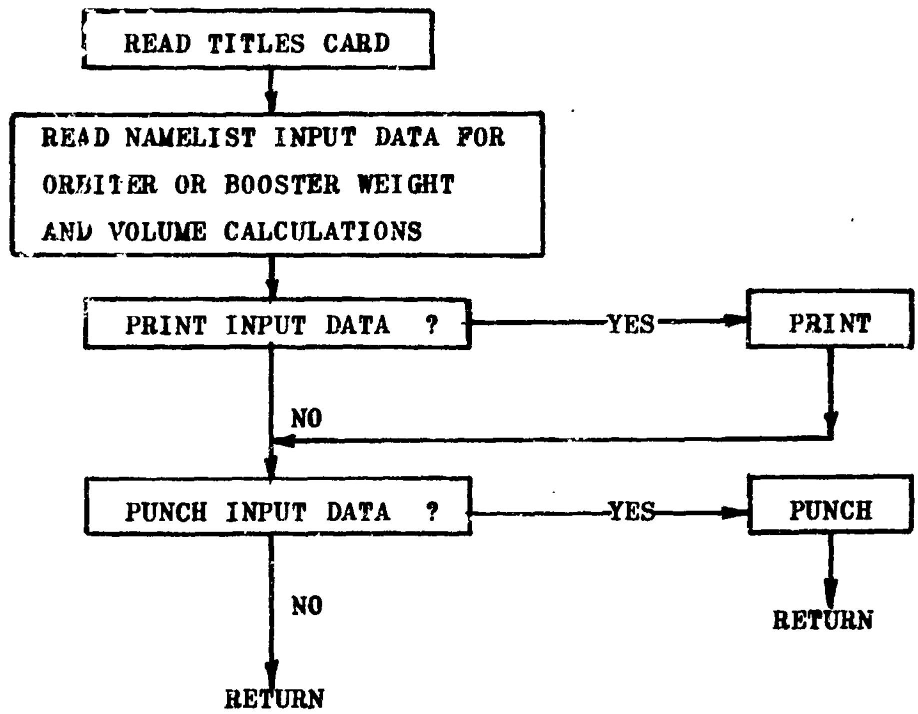


Figure 3-9 Subroutine READY Flow Diagram

VEHDF SUBROUTINE FLOWCHART

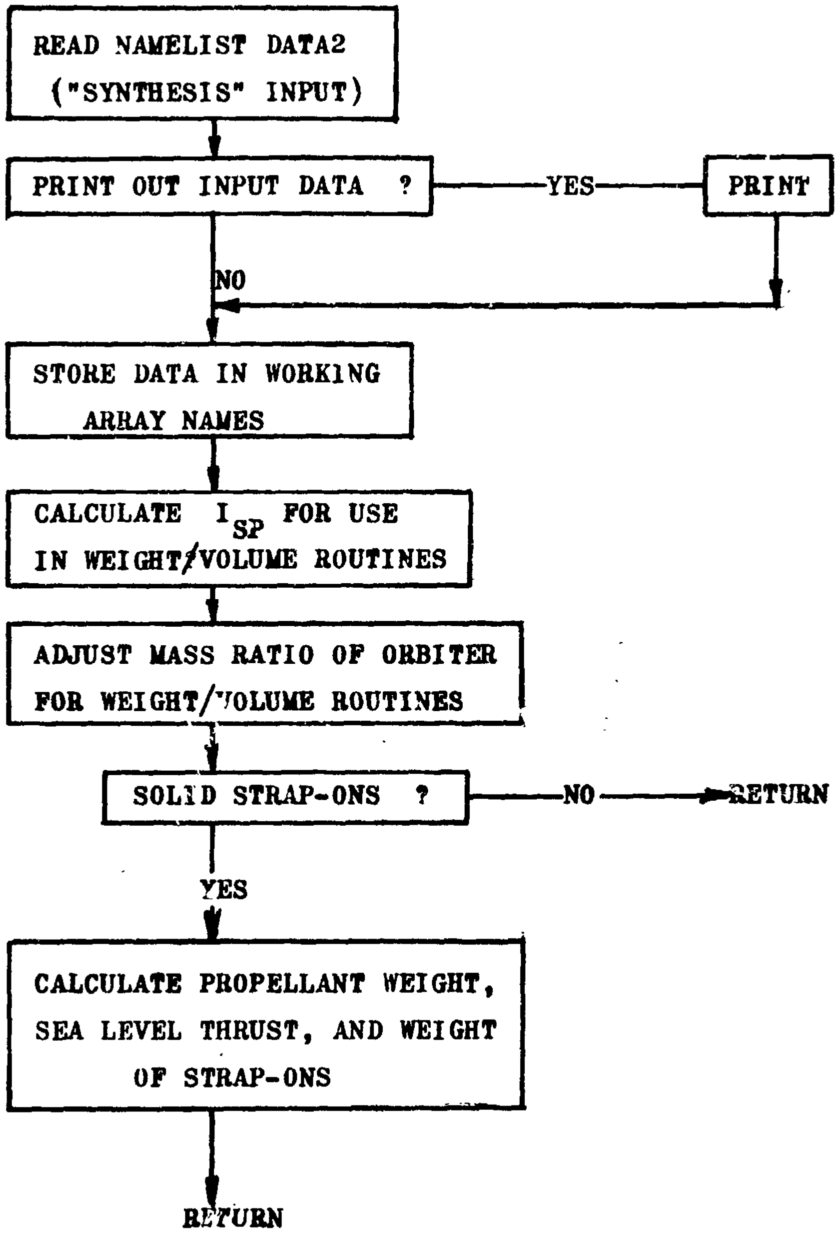


FIGURE 3-10 Subroutine VEHDF Flow Diagram

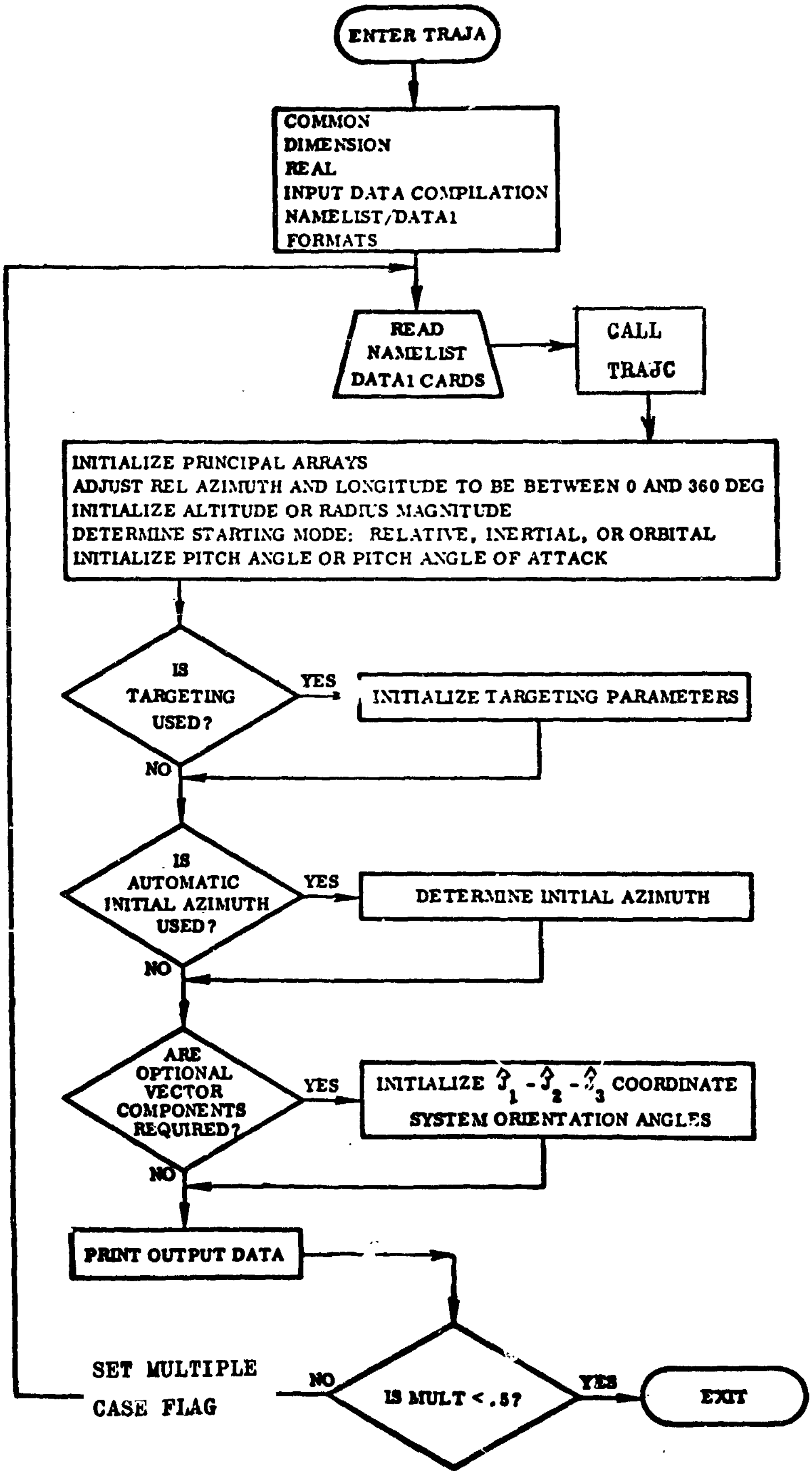


Figure 3-11. Subroutine TRAJA Flow Diagram

A dual system of defining input data is used to reduce the card input requirements and simplify input generally. Input is accomplished with the input acronyms that are defined in Section 4, Program Operation Instructions. The values of these acronyms are obtained directly from either compiled-in values or from NAMELIST input. With the exception of the table data arrays †, all input acronym values are complied-in by means of DATA statements; these compiled-in values are then subject to change if the input acronym is input with a card via NAMELIST. For simplicity there is only one NAMELIST, and if either the compiled-in value of the input acronym will suffice or if the input acronym is not required for the particular trajectory simulation, then it is not necessary to input the acronym by card via the DATA1 NAMELIST. In the case of the table data arrays which have no compiled-in values, it is always necessary to input by card those elements of the arrays which are to be used in the table. It is not necessary to input sections of an array which are not used. These elements which must be defined are specified by the input parameter which specified the number of arguments in the table. All input acronyms are floating point; those beginning with a fixed point alphabetic designator (I, J, K, L, M, or N) are defined to be floating point with a REAL statement.

A dual system of initial parameters and sectional parameters is used in GTSM. These parameters which are elements of the V, VQ, and Q arrays are initially defined by the input acronyms. For quantitative parameters this definition is usually an identify, while for some integer parameters such as flags the value is incremented internally by 0.1 to avoid truncation to the preceding integer during fixed point conversion. An example: 9 might be read as 8.999... and truncated to 8. This procedure allows the use of engineer-oriented input acronyms with "natural" values while internally taking advantage of the characteristics of the large subscripted arrays, and results in providing a permanent set of input acronym values which can only be changed by specific card input via DATA1 at the initiation of subsequent multiple cases. The V, VQ, and/or Q parameters can be redefined either by special subsequent initialization or when used as control variables in a general iteration block. The one exception to the permanence (unchanged by program operation) of input acronyms is the parameter "MULT". "MULT" has a compiled-in value of zero. Each time a multiple case is required, "MULT" must be specifically input by card in the preceding case with a value greater than 0.5. Upon completion of any case, if "MULT" is greater than 0.5 the subsequent case is initiated after "MULT" is reset to zero; if "MULT" is zero, control is returned to MAIN.

Special initialization of position, velocity, and attitude parameters is accomplished subsequent to the initial initialization of the V, VQ, and Q arrays. These initializations are required to provide flexibility in the choice of parameters from which to initiate the trajectory and include:

† These arrays are appropriately defined in Section 4 (Program Operating Instructions)

a. Specification or no specification of cycle constraints for relative azimuth and longitude. This refers to restriction, or no restriction of azimuth and longitude to primary values.

b. Specification of initial altitude or intial radius.

c. Specification of relative, inertial, or orbital starting modes.

d. Specification of initial pitch angle or initial pitch angle of attack.

e. Definition of initial parameters for down range computations.

f. Definition of initial parameters for targeting, if required.

g. Specification of initial azimuth for targeting, if required.

h. Definition of orientation parameters for optional vector components, if required.

The input data is then printed according to the following grouping:

a. Initial Conditions

b. Targeting Data

c. General Iteration Blocks

d. Section Data

e. Atmosphere Table, unless suppressed by program input

f. Aerodynamic Tables, unless suppressed by program input

g. Propulsion Tables, unless suppressed by program input

h. Pitch Rate Tables, unless suppressed by program input.

3.2.2.7 TRAJC Subroutine. The subroutine TRAJC was created to prevent the user from erroneously inputting certain program flags for the trajectory portion of the synthesis simulation. This routine is called from the subroutine TRAJA (sec. 3.2.2.6) immediately following the reading of the trajectory NAMELIST input data (sec. 4.2.3) so that the program flags set by TRAJC will override those the user inputs. The user is referred to the Compiled-In Data (Appendix V) for a complete list of these variables which may be categorized as:

- program control flags (STGC, STGT, STGD) for simulation sections 2 thru 5, and 7, and SEC)
- propulsion control parameters (TWC) for sections 1 thru 7
- iteration control (GIS, CVC1, CVN1, DCV1, ECC1, ECC2, ECN1, ECN2)
- vehicle state (PLD)
- vehicle attitude (ALPC, SALP)

For solid strap-on rockets, the first section Control Variables (STGC, STGT, STGD, and STGV) and JETW are defined.

A summary flow chart is shown in Figure 3-12 and a FORTRAN listing is presented in Appendix IV-10.

3.2.2.8 ITER8 Subroutine. The subroutine ITER8 tests the overall synthesis loop for convergence. The orbiter mass ratio used versus a basis of calculations in the weight/volume routines is compared to the orbiter mass ratio obtained from the trajectory simulation for successful mission completion. If the two values are not within the specified input tolerance, the value calculated in the trajectory simulation is used for the weight/volume calculations and the iteration continues. Upon successful convergence, the weight/volume and trajectory calculations are repeated once for print purposes. ITER8 calls RANGE to calculate the booster flyback range; and SUMOUT for printing of the Synthesis Summary Output. A flow chart of ITER8 is shown in Figure 3-13 and the FORTRAN listing appears in Appendix IV-11.

3.2.2.9 RANGE Subroutine. Five options are available to the user for calculation of the booster flyback range by the Subroutine RANGE:

1. Detailed flyback range calculated from built-in tables of booster staging conditions (altitude, flight path angle, velocity), apogee altitude and velocity, and bank angle.

2. Flyback range as a function of dynamic pressure at staging.

3. Flyback range set to an input constant.

TRAJC SUBROUTINE FLOWCHART

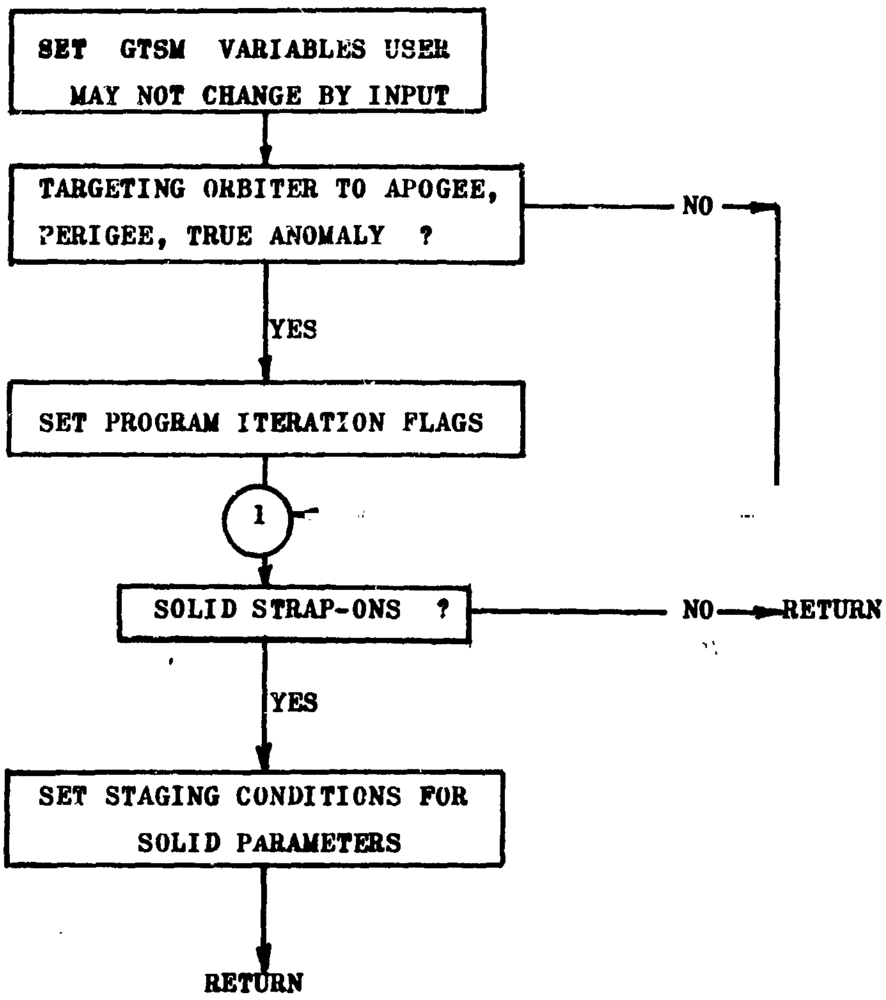


Figure 3-12 Subroutine TRAJC Flow Diagram

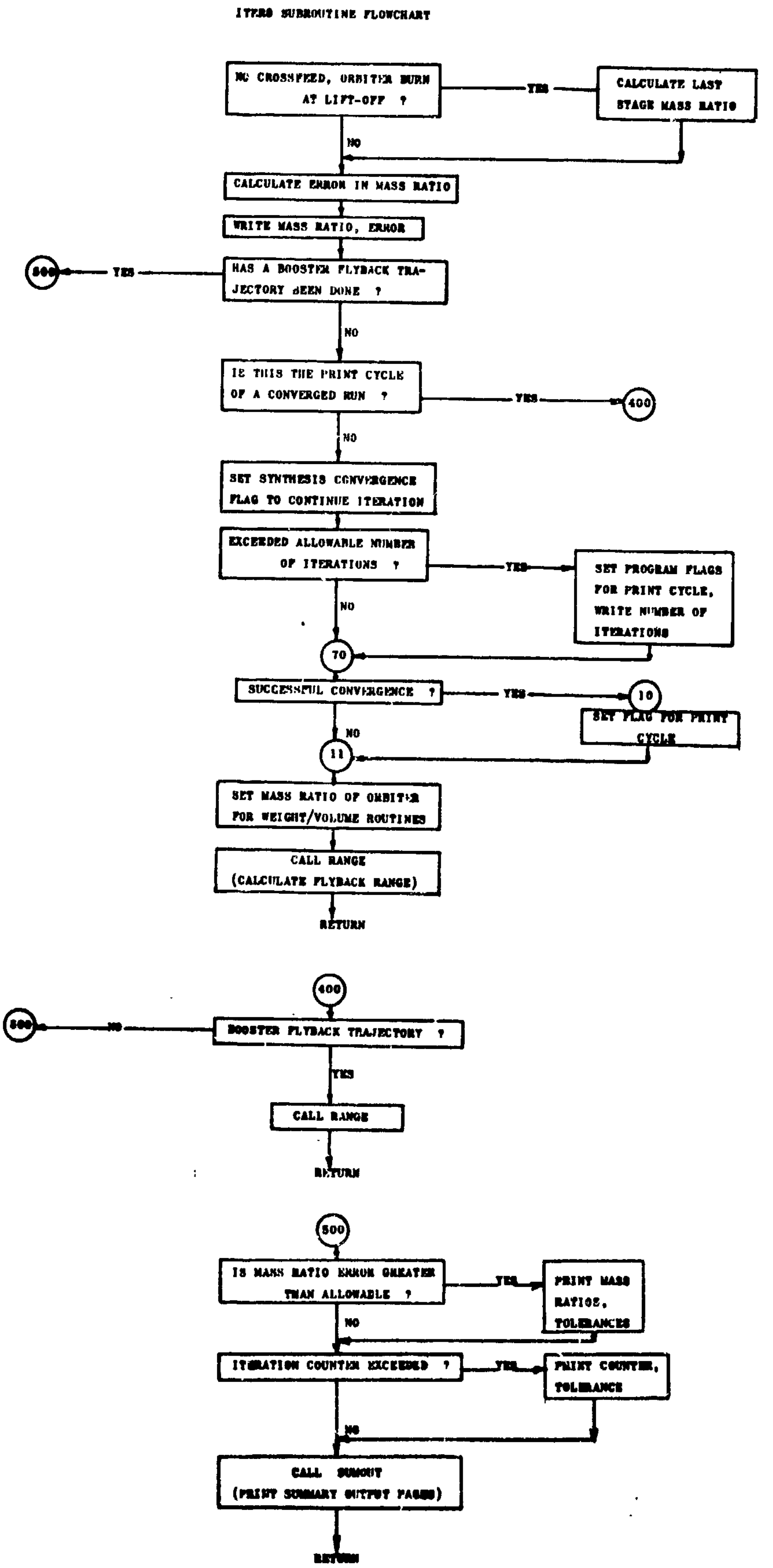


Figure 3-13 Subroutine ITER8 Flow Diagram

4. Flyback range as a function of IIP central angle.

5. Flyback range calculated by trajectory simulation.

The flowchart for RANGE is presented in Figure 3-14 and Appendix IV-12 contains the FORTRAN listing.

3.2.2.10 TBL2D Subroutine. The subroutine TBL2D is used for two dimensional interpolation by the subroutine RANGE (sec. 3.2.2.9). For two independent variables (x and y), the value of the corresponding dependent variable (z) is located. The elements of the x and y vectors must be monotonic. The routine is used by the statement

CALL TBL2D (X,Y,XTBL,YTBL,ZTBL,Z)

where

X is the value of one independent variable

Y is the value of the other independent variable

XTBL is the x vector

YTBL is the matrix of the dependent variable

Z is the value of the output dependent variable.

Z is the only output.

A flow diagram for TBL2D is shown in Figure 3-15 and the FORTRAN listing appears in Appednix IV-13.

3.2.2.11 SUMOUT Subroutine. The subroutine SUMOUT is called from ITER8 (sec. 3.2.2.3) after calculating and printing the data for a converged synthesis case. For sea level and vacuum conditions, nominal and uprated thrusts are determined for the booster and orbiter for print purposes. This routine then prints synthesis summary outputs weights, volumes, propulsion and performance data as shown in the sample output (Figures 5-8 and 5-9). Multiple copies of this data sheet may be obtained by inputting the desired number of copies as COPIES in $DATA2 (see Sec. 4). A flow diagram for SUMOUT is shown in Figure 3-16 and the FORTRAN listing is Appendix IV-14.

RANGE SUBROUTINE FLOWCHART

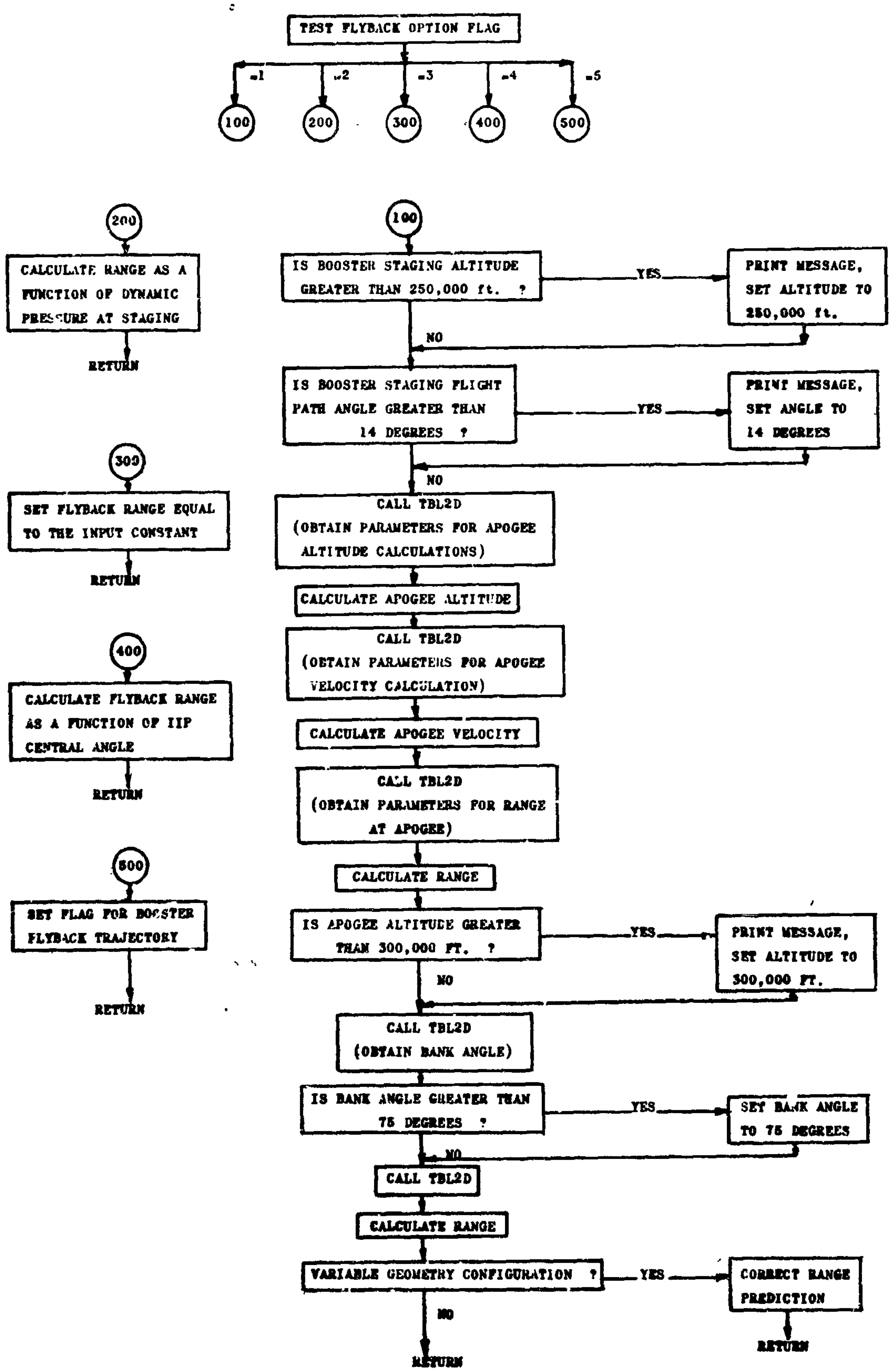


Figure 3-14 Subroutine RANGE Flow Diagram

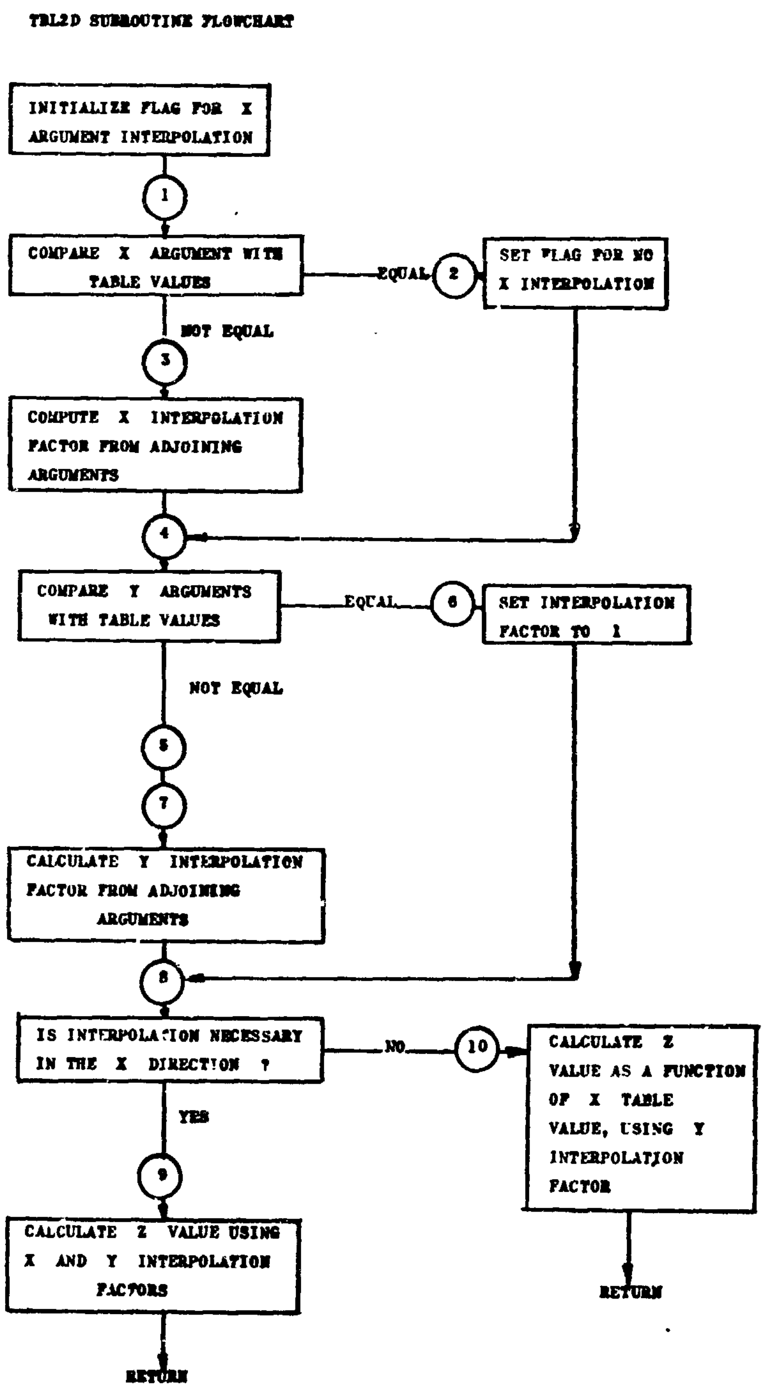


Figure 3-15 Subroutine TBL2D Flow Diagram

SUMOUT SUBROUTINE FLOWCHART

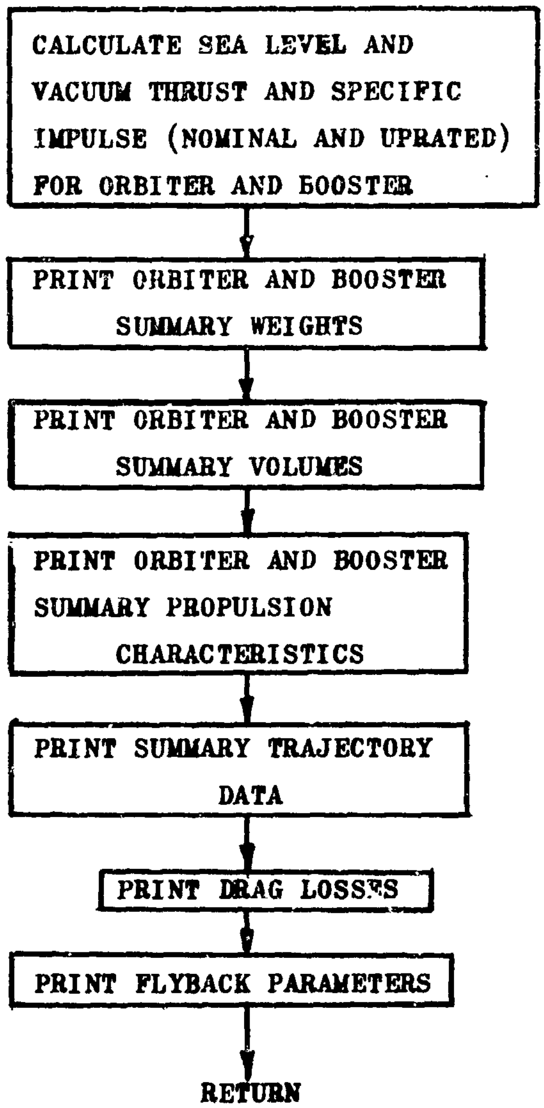


Figure 3-16 Subroutine SUMOUT Flow Diagram

3.2.3 GTSM OVERLAY

The GTSM overlay is a primary overlay comprised of its driving program (GTSM) and 19 subroutines. A detailed trajectory simulation of the booster and the orbiter is computed by this set of routines and of the booster entry if utilized.

A rigid system of nomenclature was developed for GTSM which greatly simplifies the design and coding of any option having general application to the principal groups of parameters which are required for or defined during a trajectory simulation. This system utilizes parameter arrays, which group parameters according to their use and operation in the program. The principal arrays are:

V Array. The V array consists of all principal vehicle, environmental, and trajectory parameters which are evaluated once before the initiation of each trajectory. Also included in the V array are important flags. The V array consequently defines the initial conditions and the ground rules for each trajectory simulation. The only exception to this definition is the group of iteration block parameters. These parameters are defined in both the V array and the VQ array and are subject to change (internally) whenever a new iteration block is entered.

VQ Array. The VQ array consists of all the principal parameters which are used to define iteration blocks. These parameters, known as the iteration parameters, are defined for each iteration block once before the initiation of each trajectory simulation.

Q Array. The Q array consists of all the principal vehicle modeling and trajectory control parameters together with the important flags which are required to define the simulation section. These are the simulation section conditions and are defined for each simulation section once before the initiation of each trajectory simulation.

Z Array. The Z array consists of all major vehicle, environmental, and trajectory parameters which are determined at least once for each integration step during the trajectory simulation. These parameters are known as the output parameters.

W Array. The W array consists of parameters which are evaluated from once to several times during a trajectory simulation, or are used as major flags requiring transmittal from one routine to another.

WQ Array. The WQ array consists of the subgroup of parameters contained in the W array which are used in conjunction with iteration blocks or trajectory blocks.

3.2.3.1 GTSM Program. The purpose of subroutine GTSM is to define the iterative and non-iterative portions of the trajectory simulation and then to cause the required iterations and/or simulation of trajectory segments to be initiated. A summary flow diagram of this subroutine is shown in Figure 3-17 and the corresponding FORTRAN listing is presented in Appendix IV-15.

GTSM was designed to avoid any unnecessary repetition of simulation sections during an iteration; that is the GIS blocks are automatically defined to include only the simulation sections which are required for the iteration. These simulation sections are those which contain primary * control variables, iteration end conditions, and all simulation sections in between. Sections which contain secondary* control variables are not required to be contained within their corresponding GIS block, consequently a GIS block may be used to iterate on the value of a secondary control variable which will be used subsequent to the iteration block if and only if the secondary control variable in question has a like primary control variable within the GIS block. An example of the use of this feature is the use of the pitch attitude angle as a primary control variable to obtain a specified flight path angle (iteration end condition); and also as a secondary control variable in a simulation section, subsequent to the one which contains the specified flight path angle, to automatically provide a good initial estimate for the value of a primary control variable of a subsequent iteration block. With this one exception of secondary control variables, there can be no overlapping of contained simulation sections for different GIS blocks; that is a simulation section contained on one GIS block cannot be contained in another.

Simulation sections which are contained within an iteration block must be executed once during each required iteration step** plus an additional time upon completion of the iteration, while non-iterative simulation sections are processed only once. This procedure was adopted to allow the use of short summary printout

*See Section 4.2.3.2 for the definition of primary and secondary control variables.

** An iteration step occurs when the iteration end condition(s) is(are) evaluated.

during iteration while providing detailed trajectory printout upon completion of an iteration block and results in eliminating redundant simulation section computations.

Up to four GIS iteration blocks may be used per trajectory; there are seven iteration options per GIS block. Upon definition of the simulation sections contained within a GIS block, the required iteration option is selected by means of a code flag ISLCT. These seven iteration modes provide a means to nest any permutation of iterations where the number of primary control variable-iteration end condition pairs does not exceed three.

3.2.3.2 CNTVR. The CNTVR subroutine is a storage and recall subroutine which provides the means to define the initial control variable values from the V and Q arrays. The CNTVR flow diagram is presented in Figure 3-18 and the corresponding FORTRAN listing appears in Appendix IV-16.

CNTVR has four principal functions:

a. Determination of the initial control variable values from the V and/or Q arrays and subsequent storage of these values.

b. Reset of the appropriate V and/or Q parameter(s) to the current value of the control variable(s).

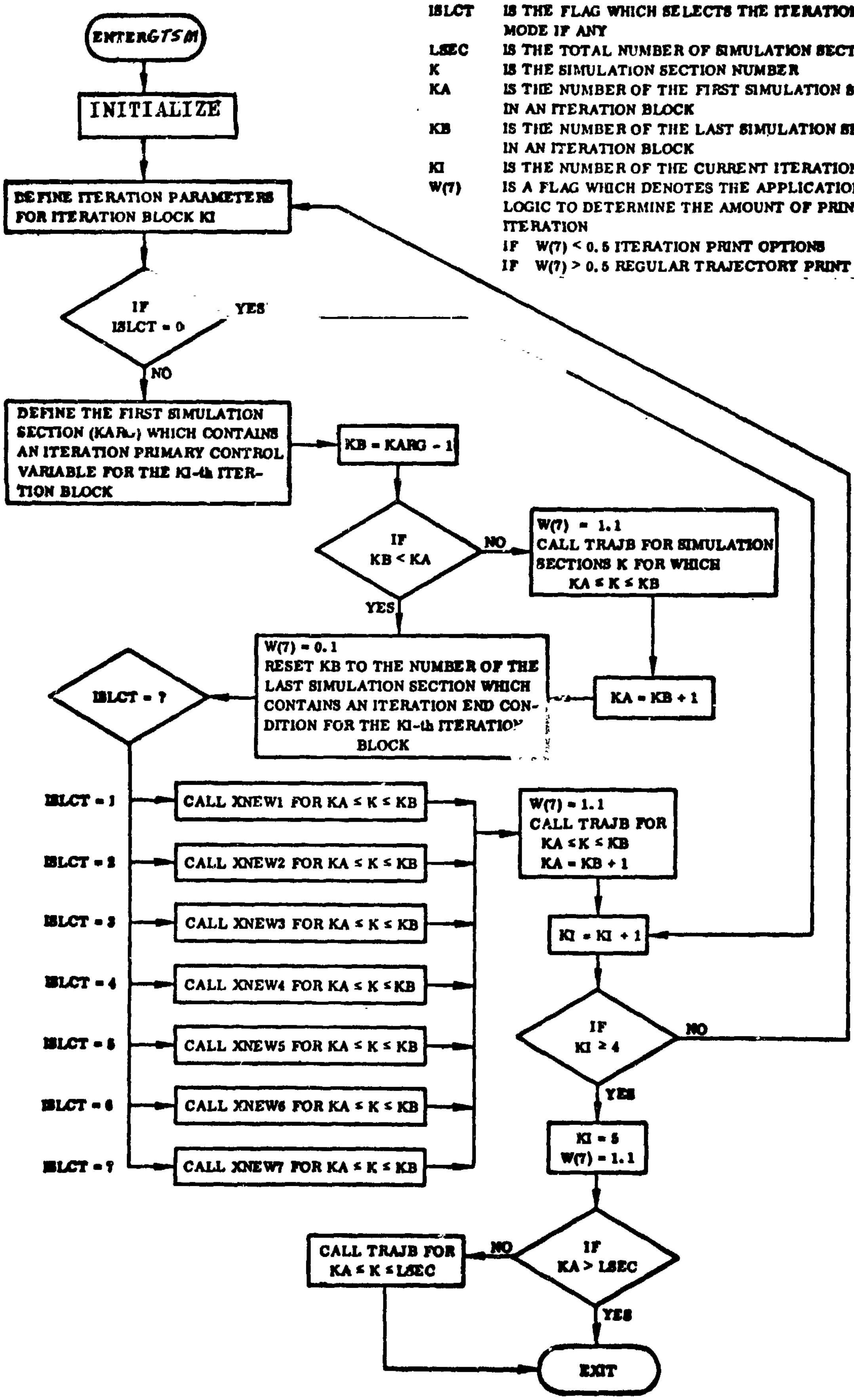
c. Reset of the appropriate V and/or Q parameter(s) to the corresponding initial value which was initially stored.

d. Re-initialization of the control variable values from the V and/or Q arrays.

The use of the large subscripted arrays V and Q provides great flexibility and simplicity for carrying out these operations.

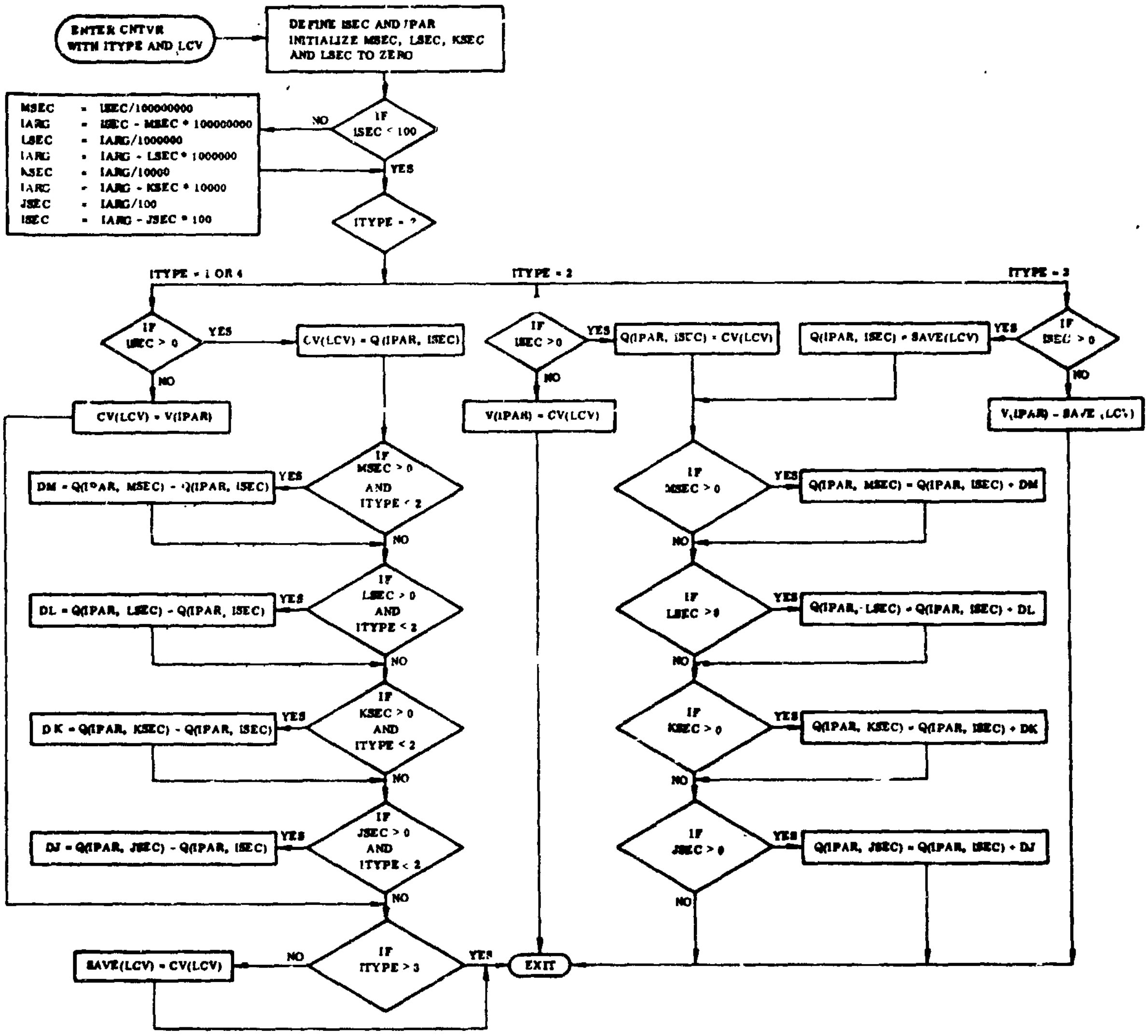
3.2.3.3 XNEW1. The XNEW1 subroutine performs a Newton-Raphson iteration for a single control variable group*-iteration end condition. The flow diagram and FORTRAN listing for the XNEW1 subroutine is presented in Figure 3-19 and Appendix IV-17. XNEW1 defines the end condition directly from a single simulation of the trajectory segment contained within the current GIS block† and is designated as a (1 × 1) iteration.

*See Section 4.2.3.2 for the definition of a control variable group.
†See Section 4.2.3.2 and 3.2.3.1 for the definition of a GIS block.



ISLCT	IS THE FLAG WHICH SELECTS THE ITERATION MODE IF ANY
LSEC	IS THE TOTAL NUMBER OF SIMULATION SECTIONS
K	IS THE SIMULATION SECTION NUMBER
KA	IS THE NUMBER OF THE FIRST SIMULATION SECTION IN AN ITERATION BLOCK
KB	IS THE NUMBER OF THE LAST SIMULATION SECTION IN AN ITERATION BLOCK
KI	IS THE NUMBER OF THE CURRENT ITERATION BLOCK
W(7)	IS A FLAG WHICH DENOTES THE APPLICATION OF LOGIC TO DETERMINE THE AMOUNT OF PRINT DURING ITERATION IF W(7) < 0.5 ITERATION PRINT OPTIONS IF W(7) > 0.5 REGULAR TRAJECTORY PRINT

Figure 3-17. Program GTSM Flow Diagram



CV(LCV)	IS THE LCV-th PRIMARY ITERATION CONTROL VARIABLE
ISEC	INITIALLY DEFINES ALL THE SIMULATION SECTIONS IN WHICH THERE ARE CONTROL VARIABLES BELONGING TO THE LCV-th ITERATION CONTROL VARIABLE GROUP. ISEC IS SUBSEQUENTLY REDEFINED TO BE THE NUMBER OF THE SIMULATION SECTION IN WHICH THE PRIMARY CONTROL VARIABLE OF THE LCV-th ITERATION GROUP IS DEFINED.
IPAR	DENOTES CODE NUMBER OF THE Q OR V PARAMETER WHICH IS USED AS A CONTROL VARIABLE
ITYPE	DENOTES THE MODE FOR THIS SUBROUTINE ITYPE = 1 FOR INITIAL ENTRIES - DEFINES THE CONTROL VARIABLE AND SAVES THE INITIAL VALUE OF THE SPECIFIED Q OR V PARAMETER ITYPE = 2 FOR INTERMEDIATE ENTRIES - SETS THE SPECIFIED Q OR V PARAMETER TO THE VALUE OF THE CONTROL VARIABLE ITYPE = 3 RESETS THE SPECIFIED Q OR V PARAMKTER TO THE INITIALLY STORED VALUE ITYPE = 4 RESETS THE CONTROL VARIABLE TO THE VALUE OF THE SPECIFIED Q OR V PARAMETER
JSEC	THE NUMBER OF THE SIMULATION SECTION IN WHICH THE FIRST SECONDARY CONTROL VARIABLE OF THE LCV-th ITERATION GROUP IS DEFINED
KSEC	THE NUMBER OF THE SIMULATION SECTION IN WHICH THE SECOND SECONDARY CONTROL VARIABLE OF THE LCV-th ITERATION GROUP IS DEFINED
LCV	DEFINES WHICH CONTROL VARIABLE GROUP IS BEING OPERATED ON
LSEC	THE NUMBER OF THE SIMULATION SECTION IN WHICH THE THIRD SECONDARY CONTROL VARIABLE OF THE LCV-th ITERATION GROUP IS DEFINED
MSEC	THE NUMBER OF THE SIMULATION SECTION IN WHICH THE FOURTH SECONDARY CONTROL VARIABLE OF THE LCV-th ITERATION GROUP IS DEFINED
Q(IPAR, K)	THE IPAR-th Q PARAMETER FOR THE K-th SIMULATION SECTION
SAVE(LCV)	THE INITIAL VALUE OF THE PRIMARY CONTROL PARAMETER BELONGING TO THE LCV-th CONTROL VARIABLE GROUP WHICH IS SAVED FOR SUBSEQUENT USE
V(IPAR)	THE IPAR-th V PARAMETER.

Figure 3-18. Subroutine CNTVR Flow Diagram

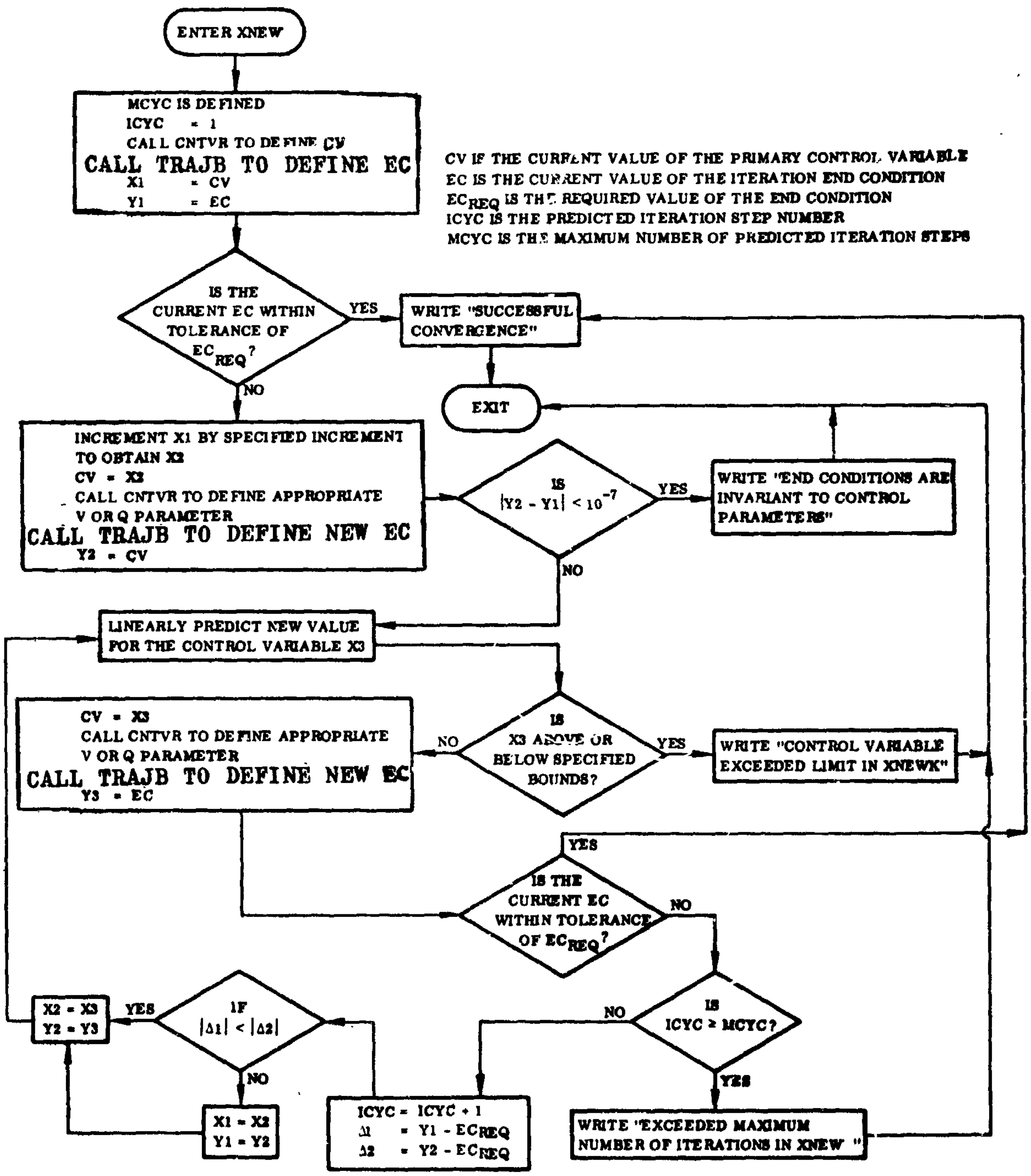


Figure 3-19. Subroutine XNEW1

The XNEW1subroutine utilizes a conventional Newton-Raphson iteration technique in which the new value of the control variable is predicted by assuming a constant slope for the function in order to predict a new "more nearly correct" value for the control variable. The first step is completed by assuming an initial value for the control variable and then computing the trajectory segment directly (V and Q arrays defined) or with internal iterations according to the iteration mode which is employed to obtain the corresponding end condition. The second step is obtained by incrementing the control variable by a specified value and in a similar manner defining the corresponding end condition. From these two sets of control variable-end condition parameter pairs, the Newton-Raphson technique is employed to predict a new control variable value. This last procedure is repeated until either successful convergence is obtained, an excessive number of iterations are made, or an error occurs. At each step, the immediately preceding control variable-end condition is used together with either the second or third preceding control variable-end condition, selected on the basis of which corresponding end condition is closest to the required end condition, to define the next value for the control variable.

3.2.3.4 XNEW4. The XNEW4 subroutine is similar to the XNEW1 subroutine in program organization and operation, however this routine is designed to perform simultaneous iterations on two control variable groups*. The flow diagram and FORTRAN listing appears in Figure 3-20 and Appendix IV-18.

The XNEW4 subroutine performs a simultaneous iteration on two control variable group-end condition pairs where the end conditions are evaluated at each iteration step by direct simulation of the trajectory segment which is contained within the GIS block† and is designated as a (2 x 2) iteration.

The program form of subroutine XNEW4 is similar to that of subroutine XNEW1 except, of course, it is expanded to process more than one iteration variable and it computes and uses iteration end condition/primary control variable partial derivatives to predict the new values for the control variables. After the initial set of control variable-end condition pairs are defined, the initial partial derivative values are evaluated. The first prediction of the values of the control variable is accomplished by using the initial values of the partial derivatives and by assuming that these values are constant over the required range of the control variable values. Prior to all subsequent iterations, the value of each partial derivative is adjusted to reflect the accuracy of the previous control variable prediction.

---

*See Section 4.2.3.2 for the definition of a control variable group.

An iteration step is completed when the iteration end conditions are evaluated.

† See Section4. 2.3.2 and 3.2.3.1for the definition of a GIS block.

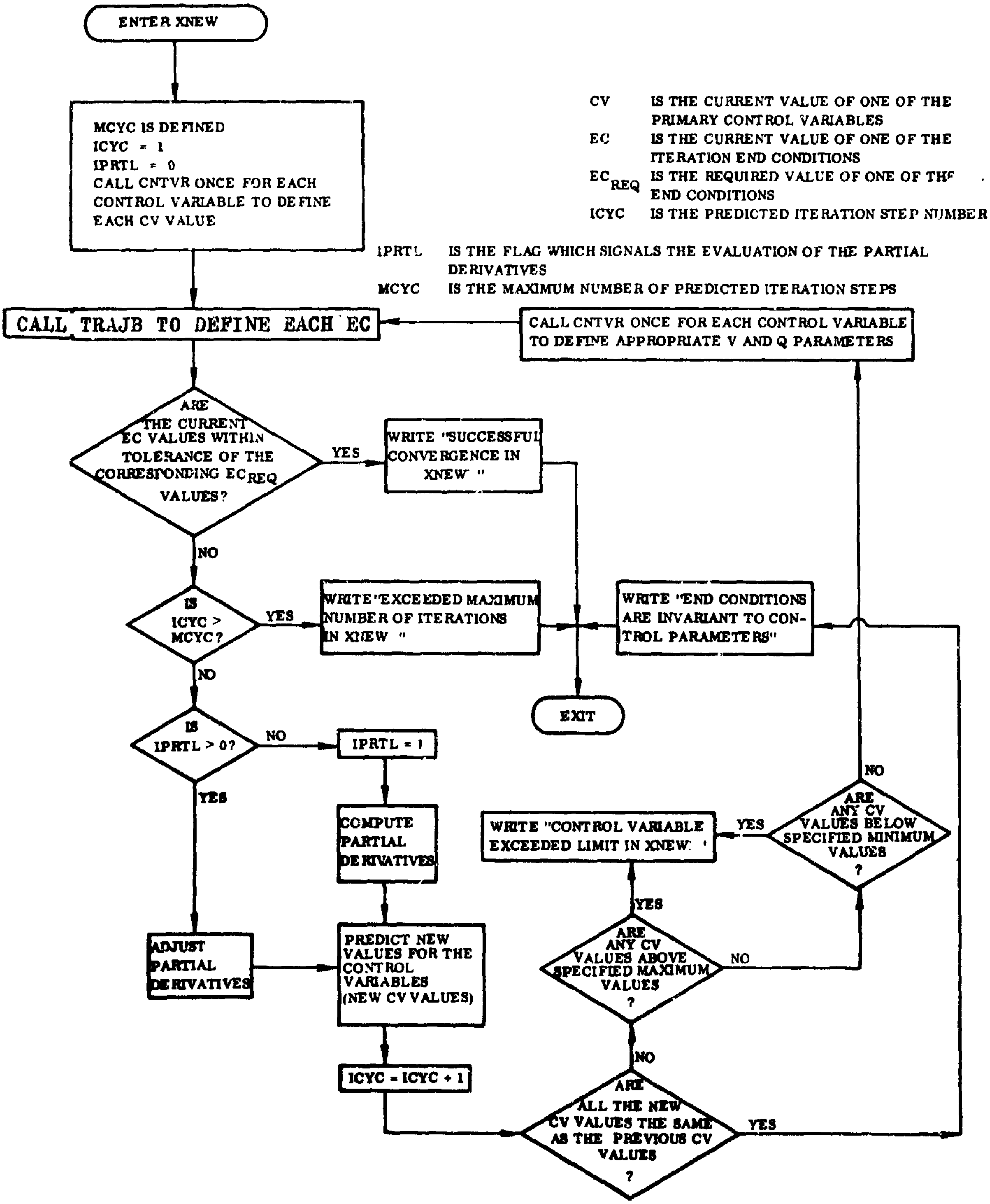


Figure 3-2Q. Subroutine XNEW4 Flow Diagram

This procedure, which was obtained from Reference 7, adjusts each partial derivative according to the following expression:

$$\left[\frac{\partial y_j}{\partial x_k}\right]_{\text{ADJUSTED}} = \left[\frac{\partial y_j}{\partial x_k}\right]_{\text{PREVIOUS}} - \left[\frac{\Delta y_j\, \Delta x_k}{\mu h_k}\right] \tag{3-1}$$

where $h_k = \Delta x_{k(I)}^2$

x_k = the kth primary control variable

y_j = the jth iteration end condition

Y_k = the required end condition value for y_k

Δx_k = the previously predicted increment for x_k

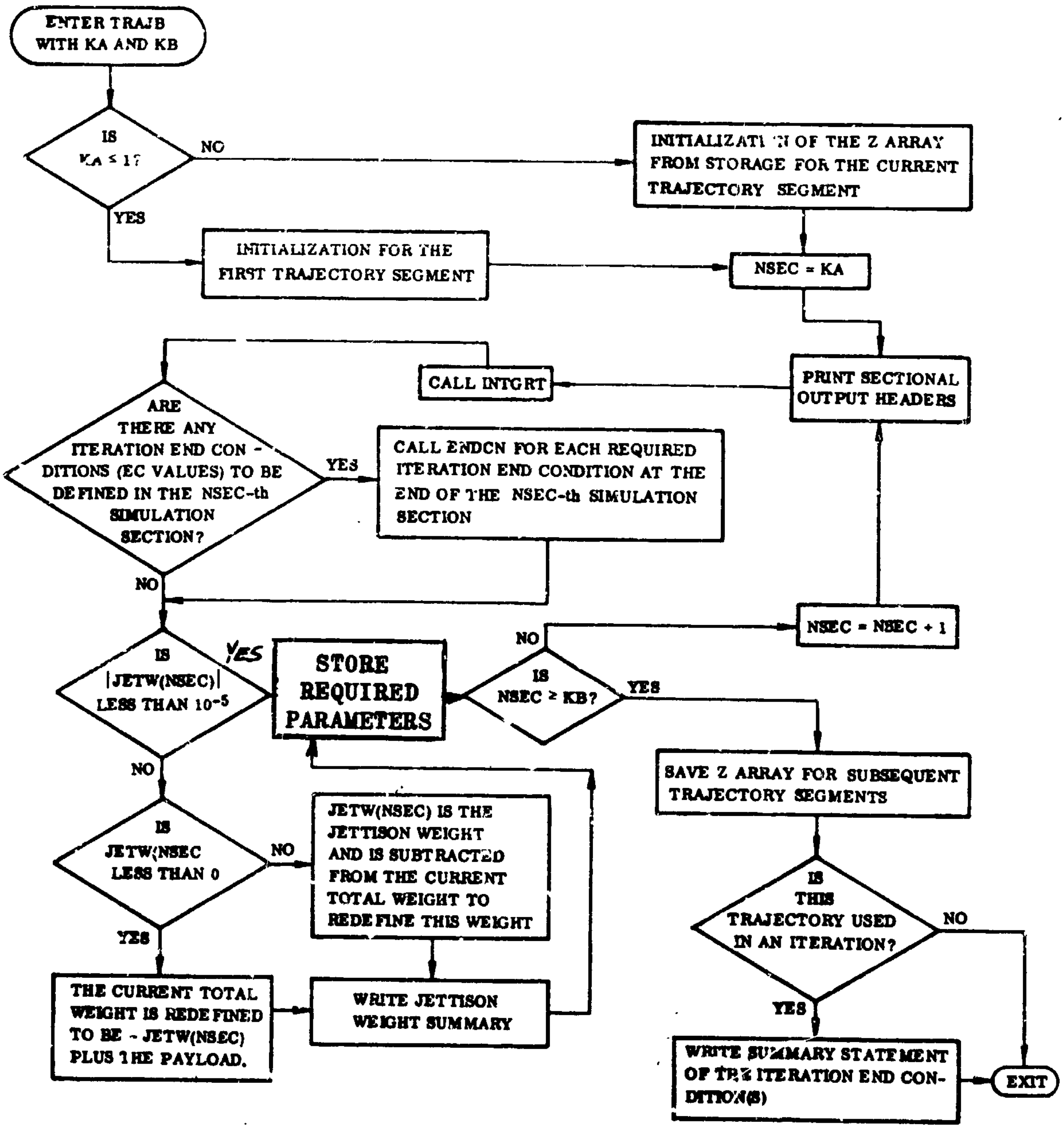
$\Delta x_{k(I)}$ = the initially specified increment for x_k

Δy_j = the last difference $Y_k - y_k$

$\mu = \sqrt{\sum_{k=1}^{N} \left(\frac{\Delta x_k^{\,2}}{h_k}\right)}$ for N primary control variables

This procedure has been used extensively and has been found to reduce the number of required iterations by about 10 to 20 percent from the conventional planar iteration procedure, which assumes constant partial derivatives.

3.2.3.5 TRAJB. The purpose of subroutine TRAJB is to initiate the numerical integration which computes the required trajectory simulation sections. TRAJB prints the simulation section output headers, the summary iteration control variable-end condition parameters statements, and the summary burnout weight-jettison weight statements. In addition the necesary bookkeeping for the initiation of a trajectory segment, for the storage of the Z array and the absolute time at the end of a trajectory segment, and for the jettison weight options is accomplished in TRAJB. A summary flow diagram is shown in Figure 3-21 and the corresponding FORTRAN



EC	IS AN ITERATION END CONDITION VALUE
JETW (NSEC)	IS THE JETTISION WEIGHT PARAMETER FOR THE NSEC-th SIMULATION SECTION
KA	IS THE NUMBER OF THE FIRST SIMULATION SECTION IN A TRAJECTORY SEGMENT
KB	IS THE NUMBER OF THE LAST SIMULATION SECTION IN A TRAJECTORY SEGMENT
NSEC	IS THE NUMBER OF THE CURRENT SIMULATION SECTION.

Figure 3-21. Subroutine TRAJB Flow Diagram

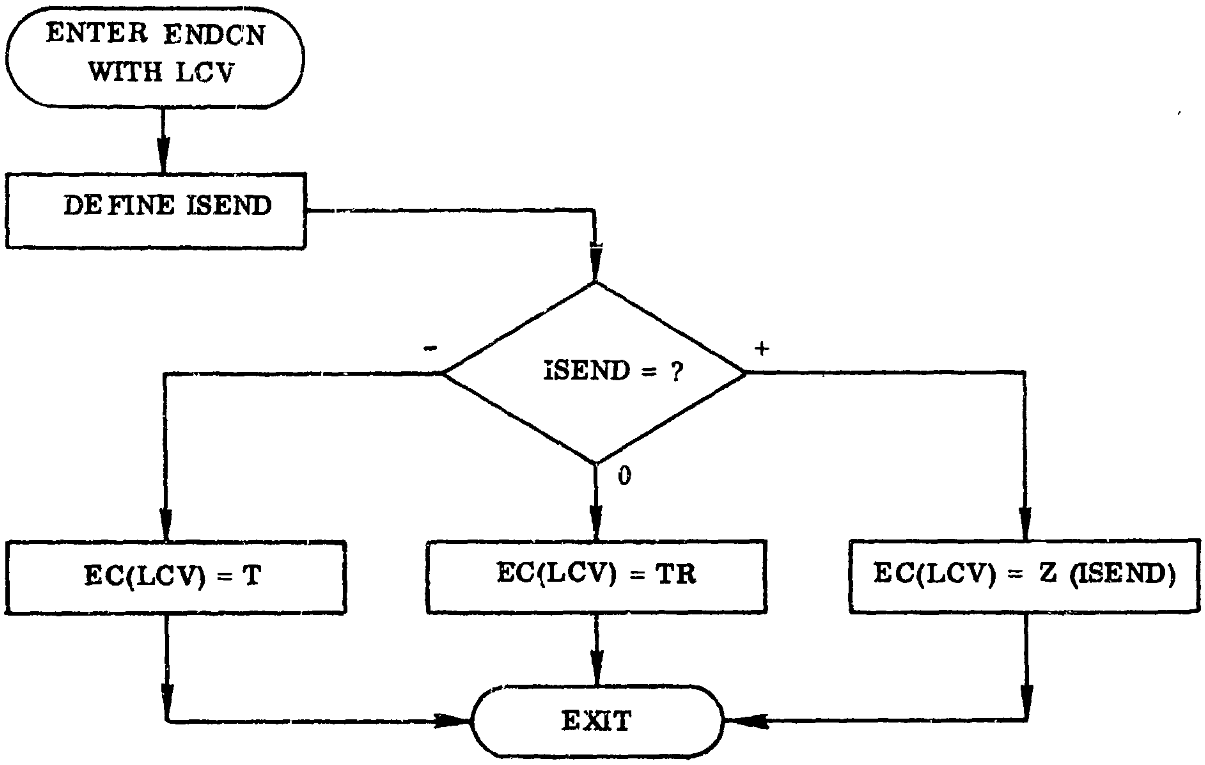
listing is presented in Appendix IV-19. TRAJB also causes the value(s) of iteration end condition(s) to be determined at the completion of their corresponding simulation section by calling subroutine ENDCN.

The jettison weight options provide the means to either subtract a specified weight from the total weight at the end of a simulation section or to reset the current total weight to a required initial weight for the subsequent simulation section. This latter option is exercised whenever the jettison weight parameter, "JETW(K)", is input with a value which is less than -0.00001; in this case the total weight is reset to a value equal to -"JETW(K)" plus the specified payload.

3.2.3.6 ENDCN. Subroutine ENDCN defines the value of the iteration end condition(s) from either the time parameters T and TR or the Z array. The simplicity of this method for defining the general iteration end conditions is a direct result of using the Z array to represent all parameters which are computed once during each integration step. A flow diagram and the FORTRAN listing of ENDCN are presented in Figure 3-22 and Appendix IV-20 respectively.

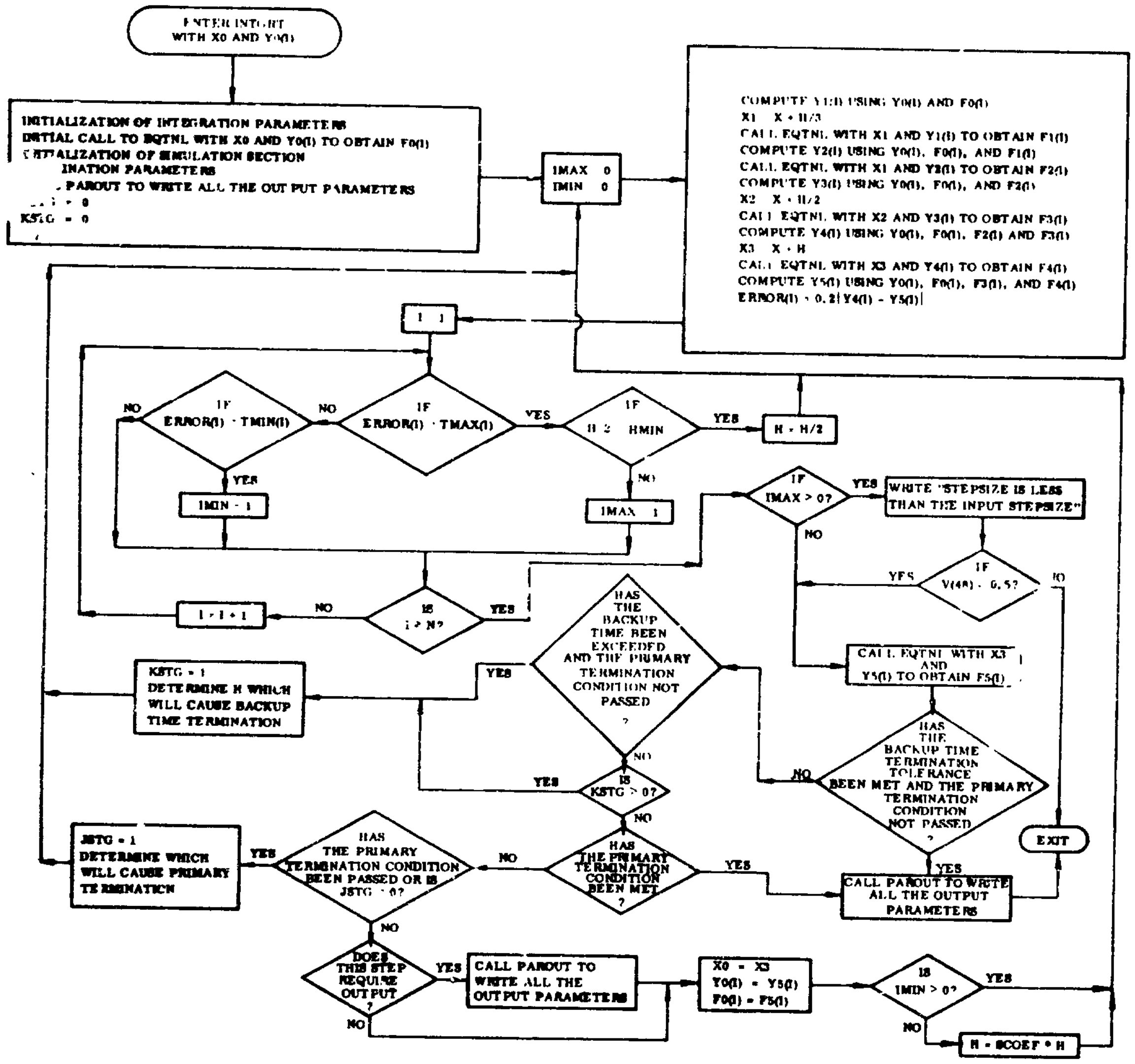
3.2.3.7 INTGRT. Subroutine INTGRT performs two functions: the first function is to numerically integrate the twelve differential equations which define the vehicle motion, weight, velocity losses, and heating parameter; the second function is to provide a general means to terminate the integration procedure. This latter function is called section termination. The flow diagram for this subroutine and the corresponding FORTRAN listing are presented in Figure 3-23 and Appendix IV-21 respectively.

The efficient yet simple Kutta-Merson numerical integration scheme, which is described in detail in Section 2.2.2.7 is used for this subroutine. This scheme is incorporated in INTGRT by simply FORTRAN coding Eq. 2-51 inside a DO loop which applies these equations to all twelve differential equations. The derivative elements (f_k) of these equations are evaluated by calling the differential equation definition subroutine EQTNL which in turn calls the necessary modeling subroutines. This evaluation can become lengthy in terms of computer time; consequently it is necessary to avoid any redundancy in this evaluation. The Kutta-Merson scheme requires at least five such evaluations per integration step. At the end of each integration step the logic tests are performed to indicate if the integration tolerances are met and whether the integration over that step is acceptable with no increase in integration step size, acceptable with a subsequent expansion in step size, or unacceptable requiring a reattempt with a smaller step size. The arithmetic operations of this integration process are performed in DOUBLE PRECISION thus eliminating uncertainties in numerical values which can occur when taking



EC(LCV)	THE LCV-th END CONDITION VALUE
LCV	DENOTES WHICH END CONDITION IS BEING DEFINEL
ISEND	CODE NUMBER FOR THE LCV-th END CONDITION
T	IS THE ABSOLUTE TIME
TR	IS THE TIME MEASURED FROM THE INITIATION OF THE CURRENT SIMULATION SECTION
Z	IS A GENERAL ELEMENT OF THE Z ARRAY.

Figure 3-22. **Subroutine ENDCN Flow Diagram**



ERROR(I)	INTEGRATION ERROR IN Y5(I)
F0(I)	INITIAL DERIVATIVE VALUE FOR THE I-th STATE VARIABLE
F1(I)	FIRST DERIVATIVE VALUE FOR THE I-th STATE VARIABLE
F2(I)	SECOND DERIVATVIE VALUE FOR THE I-th STATE VARIABLE
F3(I)	THIRD DERIVATIVE VALUE FOR THE I-th STATE VARIABLE
F4(I)	FOURTH DERIVATIVE VALUE FOR THE I-th STATE VARIABLE
F5(I)	FIFTH DERIVATIVE VALUE FOR THE I-th STATE VARIABLE
H	INTEGRATION STEP SIZE
IMAX	FLAG WHICH INDICATES NEED FOR STEP SIZE CONTRACTION BELOW THE SPECIFIED MINIMUM STEP SIZE
IMIN	FLAG WHICH INDICATES NO EXPANSION OF INTEGRATION STEP SIZE
JSTG	FLAG WHICH INDICATES THAT THE PRIMARY TERMINATION PROCEDURE HAS BEEN INITIATED
KSTG	FLAG WHICH INDICATES THAT THE BACKUP TIME TERMINATION PROCEDURE HAS BEEN INITIATED
N	NUMBER OF STATE VARIABLES
SCOEF	INTEGRATION STEP SIZE EXPANSION COEFFICIENT
TMAX(I)	UPPER INTEGRATION TOLERANCE FOR THE I-th STATE VARIABLE
TMIN(I)	LOWER INTEGRATION TOLERANCE FOR THE I-th STATE VARIABLE
V(48)	INTEGRATION MINIMUM STEP SIZE OPTION FLAG
X0	INITIAL TIME
X1	X0 + 1/3 H
X2	X0 + 1/2 H
X3	X0 + H
Y0(I)	INITIAL VALUE AT THE START OF THE INTEGRATION STEP FOR THE I-th STATE VARIABLE
Y1(I)	FIRST INTERMEDIATE VALUE OF THE I-th STATE VARIABLE
Y2(I)	SECOND INTERMEDIATE VALUE OF THE I-th STATE VARIABLE
Y3(I)	THIRD INTERMEDIATE VALUE OF THE I-th STATE VARIABLE
Y4(I)	FOURTH INTERMEDIATE VALUE OF THE I-th STATE VARIABLE
Y5(I)	FINAL VALUE AT THE END OF THE INTEGRATION STEP FOR THE I-th STATE VARIABLE

Figure 3–23. Subroutine INTGRT Flow Diagram

differences where the difference is several orders of magnitude less than the magnitude of difference variables. The Kutta-Merson integration technique as coded for INTGRT incorporates three special logic options. These are:

a. If the required step size is reduced below an input minimum allowable value, it is possible to either terminate the integration, or continue integration with the preceding integration step size which is greater than the minimum allowable step size. Both options print appropriate diagnostics indicating the integration errors which occurred.

b. Special logic is provided to reset the relative azimuth, the geocentric latitude, and longitude upon crossing a pole. This procedure is described in Section 3.2.2.5.

c. A cycle test option is provided for both the relative azimuth and the longitude. These parameters can be either constrained to the principal cycle, that is between zero and 360 degrees, or they can be allowed to take any value that results from integration. Simplicity in the use of the output data normally makes it desirable to constrain these variables to the principal cycle; however, when section termination or iteration end conditions are based on one of these parameters and it is likely that the values of the parameter will be near zero or some integral multiple of 360 degrees, it is necessary to provide continuity in the value of the parameter by not restricting it to the principal cycle. This is necessary in order to allow the appropriate iteration to operate properly.

There are two methods which are available to terminate the numerical integration; these are the primary general section termination procedure and the back-up time termination procedure.

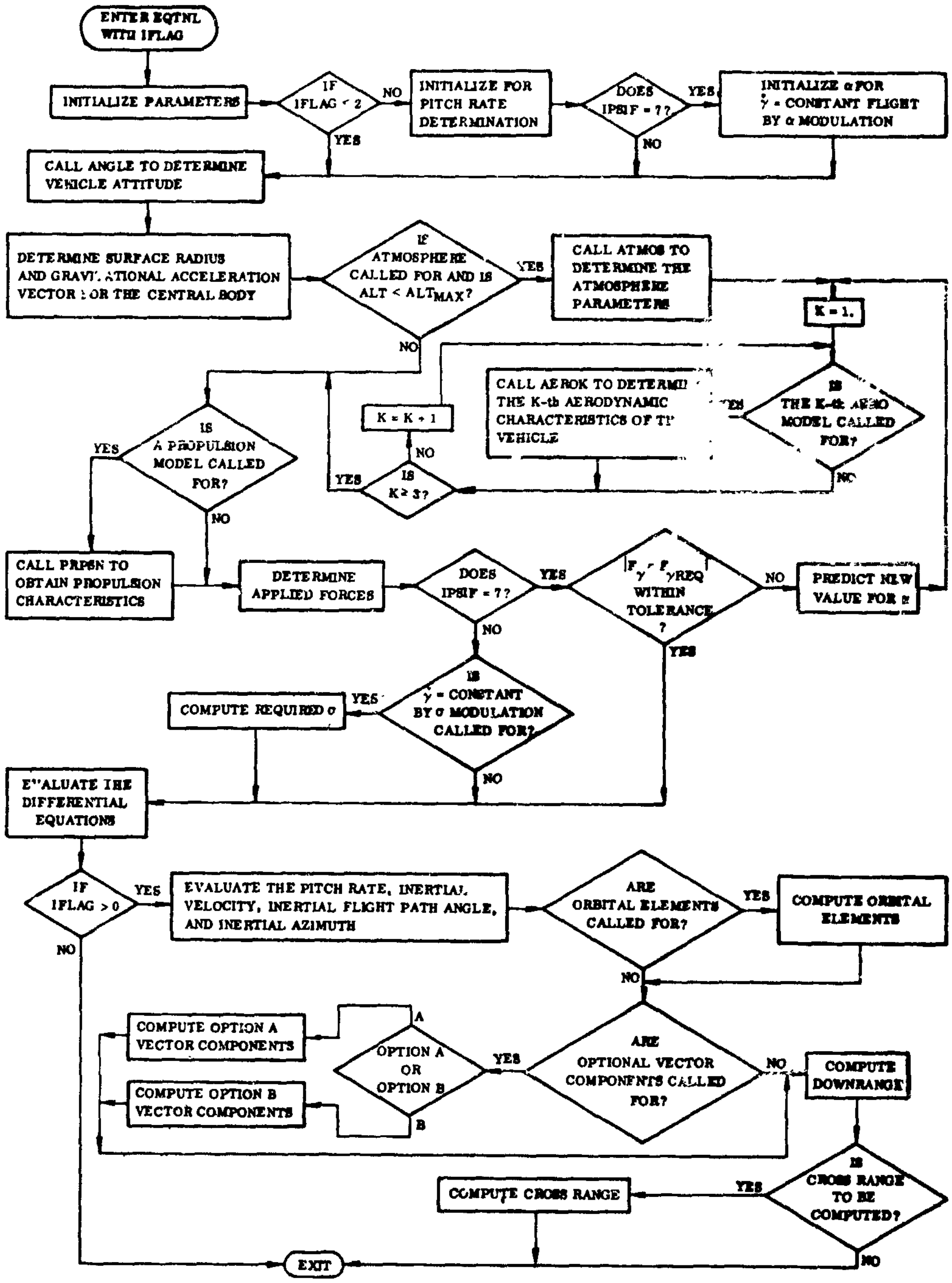
The primary termination procedure is initiated whenever the current value of the selected parameter has passed the required termination value in the specified direction. This procedure consists of using a Newton-Raphson iteration to determine the value of the integration step size which will result in the termination parameter attaining the specified termination value and then reattempting the iteration step. Up to ten such iterations are allowed for any termination; in the event that termination is not completed by the end of the tenth iteration, termination is then executed anyway with an appropriate diagnostic. This general termination procedure allows the use of the time parameters (absolute time or section relative time) or any parameter of the Z array to be used as the termination parameter.

The backup time termination procedure is an auxiliary termination option and is initiated whenever the time exceeds a specified (by program input) time value and the primary termination logic has not been entered. Either absolute time or section relative time may be used. The termination procedure is identical to that of the primary termination option except that it is secondary to the primary option and limited to the time parameters.

3.2.3.8 PAROUT. Subroutine PAROUT is used to output the values of the parameters which are computed during each integration step. These parameters which include the absolute time, the section relative time and the specified elements of the Z array are output according to the specified output option. A FORTRAN listing of PAROUT is presented in Appendix IV-22.

3.2.3.9 EQTNL. The primary purpose of subroutine EQTNL is to evaluate the derivative functions which comprise the twelve differential equations that are numerically integrated. EQTNL is also used to either directly evaluate or cause evaluation to be accomplished of all the trajectory parameters which are computed at least once during each integration step. A summary flow diagram and FORTRAN listing of subroutine EQTNL are shown in Figure 3-24 and Appendix IV-23, respectively.

To evaluate the differential equations, it is first necessary to evaluate all of the trajectory parameters which contribute to the value of the coefficients and terms of these equations. These parameters, known as the necessary parameters, define or contribute to the definition of the forces (both environmental and applied) which act on the vehicle. After the initialization of parameters which is performed on the first call to EQTNL during a simulation section, the vehicle attitude angles are defined by calling subroutine ANGLE. Next the planet surface geometry and gravitational acceleration vector are determined. The current altitude can then be determined and is used subsequently in the evaluation of the atmospheric environmental parameters if they are required. These parameters are evaluated by calling subroutine ATMOS. If applied aerodynamic force models are specified and if the atmospheric environmental model has been defined, these aerodynamic forces are defined by calling subroutines AERO1 and/or AERO2 and/or AERO3 as appropriate. Next the propulsive characteristics are determined, if they are required by calling subroutine PRPSN. At this point all the parameters which are required to define the total applied forces have been defined. If α modulation is required for constant $\dot{\gamma}$ flight (see Section 2.2.2), a Newton-Raphson iteration is used to iteratively determine the value of α to meet the conditions for constant $\dot{\gamma}$ flight. In this case the applied aerodynamic and propulsive forces must be reevaluated as they are



AEROK IS THE K-th AERODYNAMIC MODEL; AERO1, AERO2 OR AERO3
ALT IS THE CURRENT ALTITUDE
ALT_{MAX} IS THE ALTITUDE ABOVE WHICH IT IS ASSUMED THERE IS NO ATMOSPHERE
F_γ IS THE TOTAL APPLIED FORCE ACTING IN THE $\hat{i}_\gamma$ DIRECTION
$F_{\gamma_{REQ}}$ IS THE REQUIRED VALUE OF F_γ TO MAINTAIN $\dot{\gamma}$ = CONSTANT FLIGHT
IFLAG IS THE FLAG THAT DENOTES WHICH ENTRY TO EQTNL
IFLAG = 0 INTERMEDIATE ENTRIES
IFLAG = 1 LAST ENTRY
IFLAG = 2 INITIAL ENTRY
IPSIF IS THE FLAG WHICH DEFINES THE α CONTROL OPTION
α IS THE PITCH ANGLE OF ATTACK
$\dot{\gamma}$ IS THE TIME RATE OF CHANGE OF RELATIVE FLIGHT PATH ANGLE
σ IS THE ROLL (BANK) ANGLE

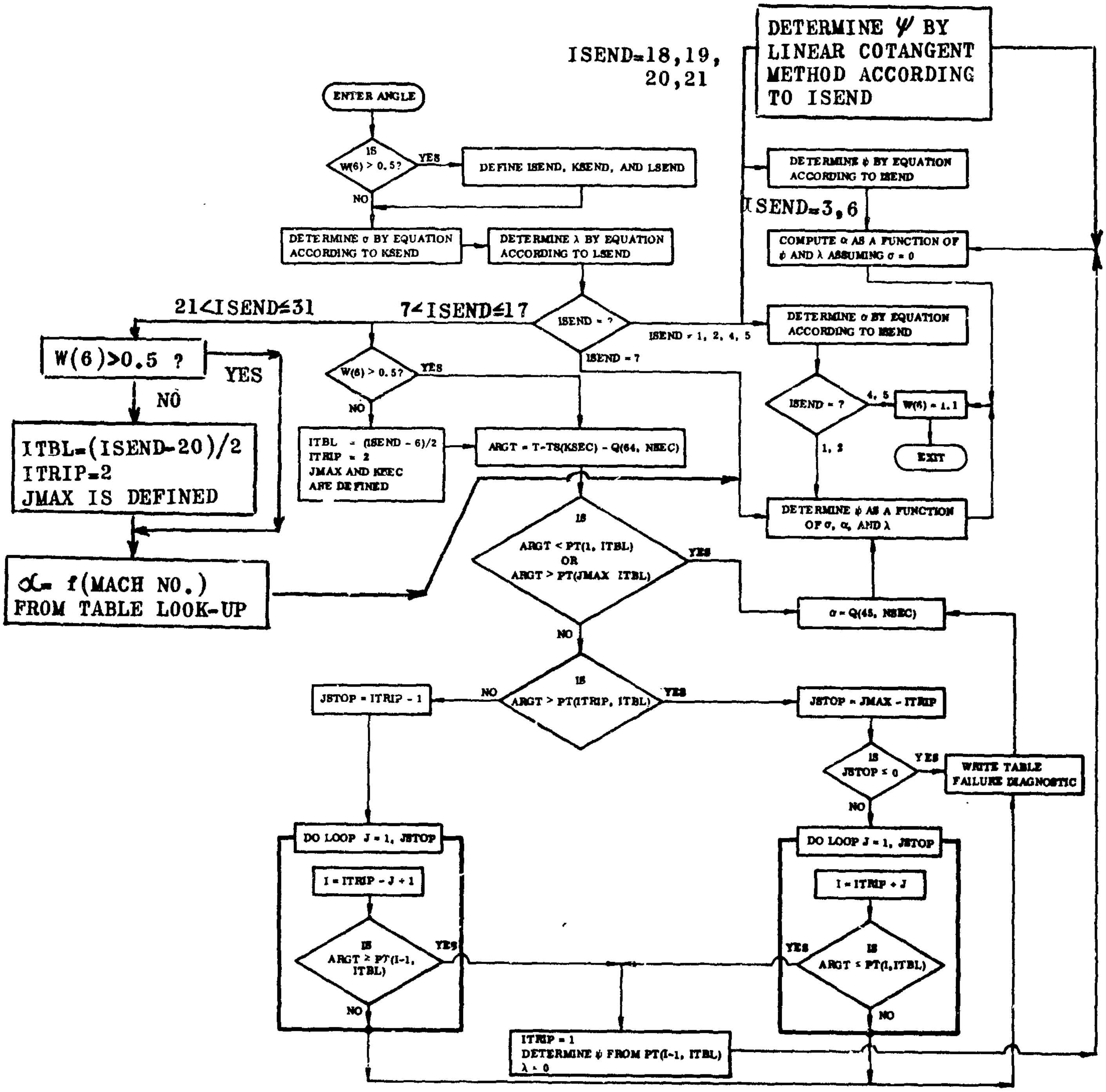
Figure 3-24. Subroutine EQTNL Flow Diagram functionally dependent on α. If σ modulation for a constant $\dot{\gamma}$ flight is required (α and σ modulation cannot be called simultaneously), the roll angle (σ) is simply adjusted to meet the constant $\dot{\gamma}$ condition; in this case it is not necessary to iterate on the value of σ. It is noted that either α or σ modulation can be used with either or both applied aerodynamics or/and propulsive forces. After the applied forces and any required α or σ modulation is computed, the differential equations are then evaluated. At this point, the computations which are required during the numerical integration process have been completed and for intermediate calls from INTGRT control is returned to INTGRT.

For the first and the last call from INTGRT and at the successful completion of an integration step, all other trajectory parameters which are computed at least once during an integration step are now evaluated. The computations of these parameters, which are called the after-the-fact parameters, are deleted for the intermediate calls to EQTNL during integration steps as these parameters are not required in the evaluation of the terms and coefficients which comprise the differential equations. The effective pitch rate (as specified or an average) is determined first; then the inertial velocity, flight path angle, and azimuth are evaluated. If the optional orbital elements are required, these are evaluated next. Instantaneous impact conditions can also be computed if required. These conditions are an optional part of the orbital elements. If the optional vector components are required, they are then determined. The downrange is then evaluated and if the cross range is required, it is subsequently determined. The radar site information (e.g., slant range, elevation angle, and azimuth) is computed as an optional part of the downrange - cross range segment of EQTNL. Upon evaluation of the downrange, control is returned to subroutine INTGRT.

3.2.3.10 ANGLE. The attitude of the vehicle is determined in subroutine ANGLE. A summary flow diagram and FORTRAN listing of subroutine ANGLE are presented in Figure 3-25 and Appendix IV-24, respectively.

The roll (bank) and yaw angle of attack are defined according to the option which is specified by program input. The option by which the pitch orientation of the vehicle is determined is also defined by program input. In this case either the pitch angle* or the pitch angle of attack can be defined first; and the other of these two parameters can then be defined. These options are described in detail in Section 4.2.3.6. The pitch angle can be defined directly by a simple linear equation or it can be obtained from a pitch rate table. For this latter case, constant pitch rates are assumed during specified time intervals. To facilitate use of this table, an equivalent pitch angle table which defines pitch angle as a function of time is defined on the initial call to ANGLE during a simulation section. The table look-up procedure then obtains

*The pitch angle is the angle from the current geocentric radius vector (up direction) to the positive vehicle roll axis.



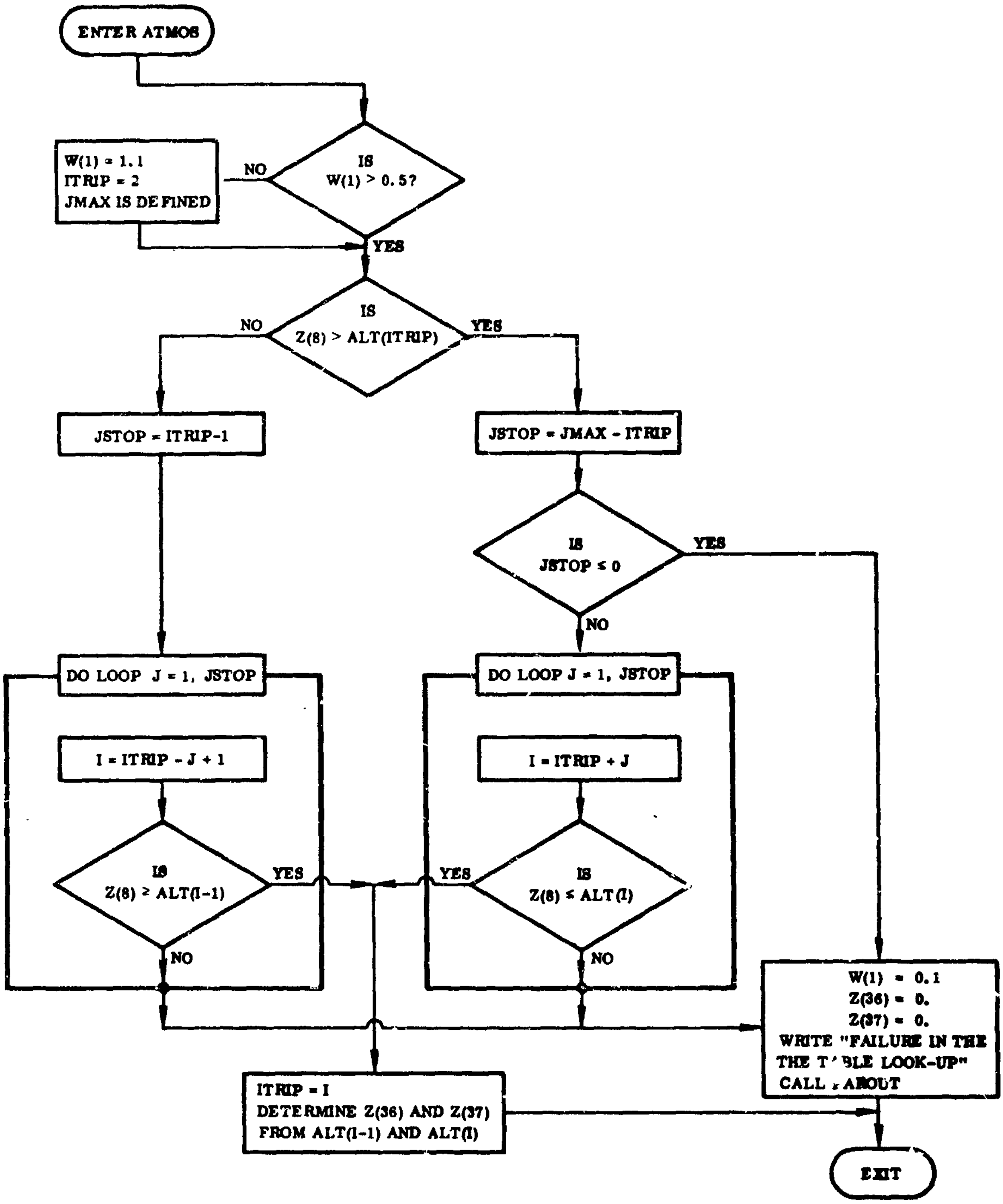
ARGT	IS THE CURRENT TIME ARGUMENT TO BE USED IN THE TABLE LOOK-UP PROCEDURE
ISEND	DEFINES WHICH OPTION IS USED TO DETERMINE EITHER σ OR ψ
ITBL	DEFINES WHICH ψ TABLE IS USED
ITRIP	IS THE MEMORY FLAG WHICH IS USED TO DETERMINE WHERE TO INITIATE THE TABLE SEARCH PROCEDURE
JMAX	IS THE TABLE SEARCH STOP PARAMETER WHICH LIMITS THE LENGTH OF THE TABLE SEARCH PROCEDURE
KSEC	IS THE NUMBER OF THE SIMULATION SECTION FROM WHICH THE TIME ARGUMENT OF THE ψ TABLE IS REFERENCED
KSEND	DEFINES WHICH OPTION IS USED TO DETERMINE σ
LSEND	DEFINES WHICH OPTION IS USED TO DETERMINE λ
NSEC	IS THE NUMBER OF THE CURRENT SIMULATION SECTION
PT(I,ITBL)	IS THE I-th INPUT TIME ARGUMENT FOR THE ITBL-th ψ TABLE
Q(45,NSEC)	IS A CONSTANT PITCH ANGLE OF ATTACK TERM FOR THE NSEC-th SIMULATION SECTION
Q(64,NSEC)	IS A CONSTANT TIME TERM WHICH IS USED WHEN DEFINING ARGT FOR THE NSEC-th SIMULATION SECTION
T	IS THE CURRENT ABSOLUTE TIME
TS(KSEC)	IS THE ABSOLUTE TIME AT THE INITIATION OF THE KSEC-th SIMULATION SECTION
W(6)	IS THE INITIALIZATION FLAG W(6) = 0.1 FOR THE INITIAL CALL TO ANGLE DURING A SIMULATION SECTION W(6) = 1.1 FOR SUBSEQUENT CALLS TO ANGLE
α	IS THE PITCH ANGLE OF ATTACK
λ	IS THE YAW ANGLE OF ATTACK
σ	IS THE ROLL (BANK) ANGLE
ψ	IS THE PITCH ANGLE

Figure 3-25. Subroutine ANGLE Flow Diagram

the pitch angle at the initiation of the current pitch rate and then adds it to the change in pitch angle which occurs from this time. The pitch angle table is provided with a memory flag which after the initial call to ANGLE (at the beginning of a simulation section), starts the table search procedure at the time of the previous entry. This feature minimizes the search time and yields an extremely fast table look-up procedure. Five of these pitch rate tables are available; any one of these can be specified during a simulation section.

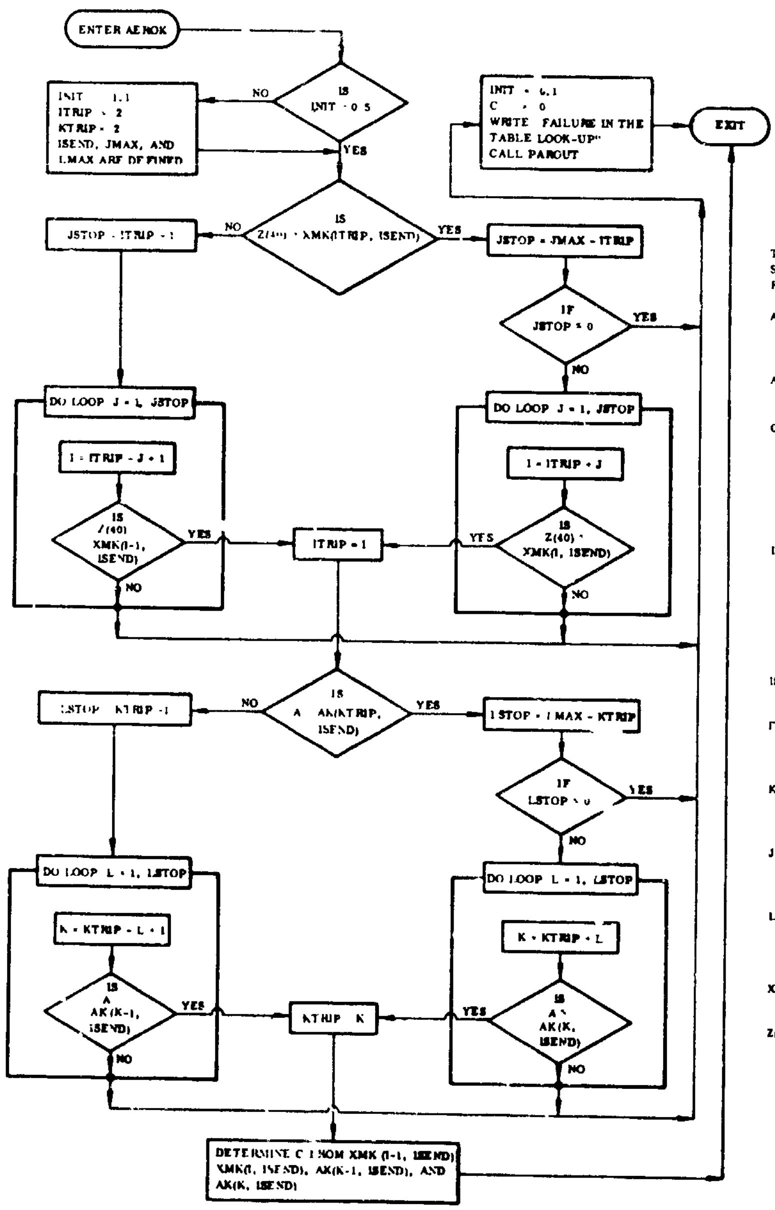
3.2.3.11 ATMOS. Subroutine ATMOS is used to define the atmospheric ambient pressure and the velocity of sound as a function of altitude. This definition is accomplished with a single independent variable (the altitude) table look-up procedure. The table look-up procedure utilizes a memory flag which starts the table search procedure at the altitude value of the previous entry for all calls to ATMOS after the initial call at the start of a simulation. This feature provides an extremely rapid evaluation of the two dependent variables, the atmospheric pressure and the velocity of sound. A summary flow diagram and corresponding FORTRAN listing are presented in Figure 3-26 and Appendix IV-25, respectively.

3.2.3.12 AERO1, AERO2, AND AERO3. Subroutines AERO1, AERO2, and AERO3, which are called the AEROK subroutines, are used to define the aerodynamic yaw force coefficient (C_{η}), the aerodynamic axial force coefficient (C_{ξ}), and the aerodynamic normal force coefficient (C_{ζ}), respectively, as a function of Mach number and the appropriate angle of attack. See Section 4.:.3.6 for the definition of coefficients and the corresponding independent variables. These evaluations are performed by means of a double independent variable (Mach number and the appropriate angle of attack) table look-up procedure similar to the single variable procedure used in subroutine ATMOS. This procedure utilizes two memory flags to initiate the table search procedure at the Mach number and angle of attack values of the previous entry for all calls to AEROK after the initial call at the start of a simulation, and consequently provides an extremely rapid definition of the required aerodynamic coefficient. A summary flow diagram and corresponding FORTRAN listing are shown in Figure 3-27 and Appendices IV-26, IV-27, and IV-28. The AEROK subroutines are identical in form with the exception of a minor logic statement which is included in AERO1 to provide automatic symmetry about a zero yaw angle of attack for C_{η}. There are five available tables for AERO2 and AERO3. The yaw modeling of AERO1 is currently not available in order to reduce core requirements. A dummy "do nothing" AERO1 is currently used in SSSP. Any one of these five tables for each AEROK subroutine may be specified during a simulation section.



ALT(I)	IS THE I-th INPUT ALTITUDE ARGUMENT FOR THE ATMOSPHERE TABLE
ITRIP	IS THE MEMORY FLAG WHICH IS USED TO DETERMINE WHERE TO INITIATE THE TABLE SEARCH PROCEDURE
JMAX	IS THE TABLE SEARCH STOP PARAMETER WHICH LIMITS THE LENGTH OF THE TABLE SEARCH PROCEDURE
W(1)	IS THE INITIALIZATION FLAG, W(1) = 0.1 FOR THE INITIAL CALL TO ATMOS DURING A SIMULATION SECTION, W(1) = 1.1 FOR THE SUBSEQUENT CALLS TO ATMOS
Z(8)	IS THE CURRENT ALTITUDE
Z(36)	IS THE CURRENT ATMOSPHERIC AMBIENT PRESSURE
Z(37)	IS THE CURRENT VELOCITY OF SOUND

Figure 3-26. Subroutine ATMOS Flow Diagram



THIS FLOW DIAGRAM IS APPLICABLE TO THE AERODYNAMIC SUBROUTINES DENOTED IN GENERAL BY "AERO". THESE SUBROUTINES ARE "AERO2" AND "AERO3".

A	CURRENT ANGLE OF ATTACK PARAMETER (Z(13))
AK	IS THE ANGLE OF ATTACK TABLE ARGUMENTS A2, A3 FOR THE AERO2, AERO3 TABLES RESPECTIVELY
C	CURRENT AERODYNAMIC FORCE COEFFICIENT FOR AERO2, C IS THE CURRENT AERODYNAMIC AXIAL FORCE COEFFICIENT (C(22)) FOR AERO3, C IS THE CURRENT AERODYNAMIC NORMAL FORCE COEFFICIENT (C(23))
INIT	IS THE INITIALIZATION FLAG INIT = 0.1 FOR THE INITIAL CALL TO AEROK DURING A SIMULATION SECTION INIT = 1.1 FOR SUBSEQUENT CALLS TO AEROK IN AERO2, INIT IS W(3) IN AERO3, INIT IS W(4)
ISEND	DEFINES WHICH OF FIVE POSSIBLE TABLES IS USED TO DEFINE C
ITRIP	IS THE MEMORY FLAG FOR THE MACH NUMBER ARGUMENTS AND IS USED TO DETERMINE WHERE TO INITIATE THE TABLE SEARCH PROCEDURE
KTRIP	IS THE MEMORY FLAG FOR THE ANGLE OF ATTACK ARGUMENTS AND IS USED TO DETERMINE WHERE TO INITIATE THE TABLE SEARCH PROCEDURE
JMAX	IS THE TABLE SEARCH STOP PARAMETER WHICH LIMITS THE LENGTH OF THE TABLE SEARCH PROCEDURE FOR THE MACH NUMBER ARGUMENTS
LMAX	IS THE TABLE SEARCH STOP PARAMETER WHICH LIMITS THE LENGTH OF THE TABLE SEARCH PROCEDURE FOR THE ANGLE OF ATTACK ARGEMENTS
XMK	IS THE MACH NUMBER TABLE ARGUMENTS XM1, XM2, XM3 FOR THE AERO1, AERO2, AERO3 TABLES RESPECTIVELY
Z(40)	CURRENT MACH NUMBER

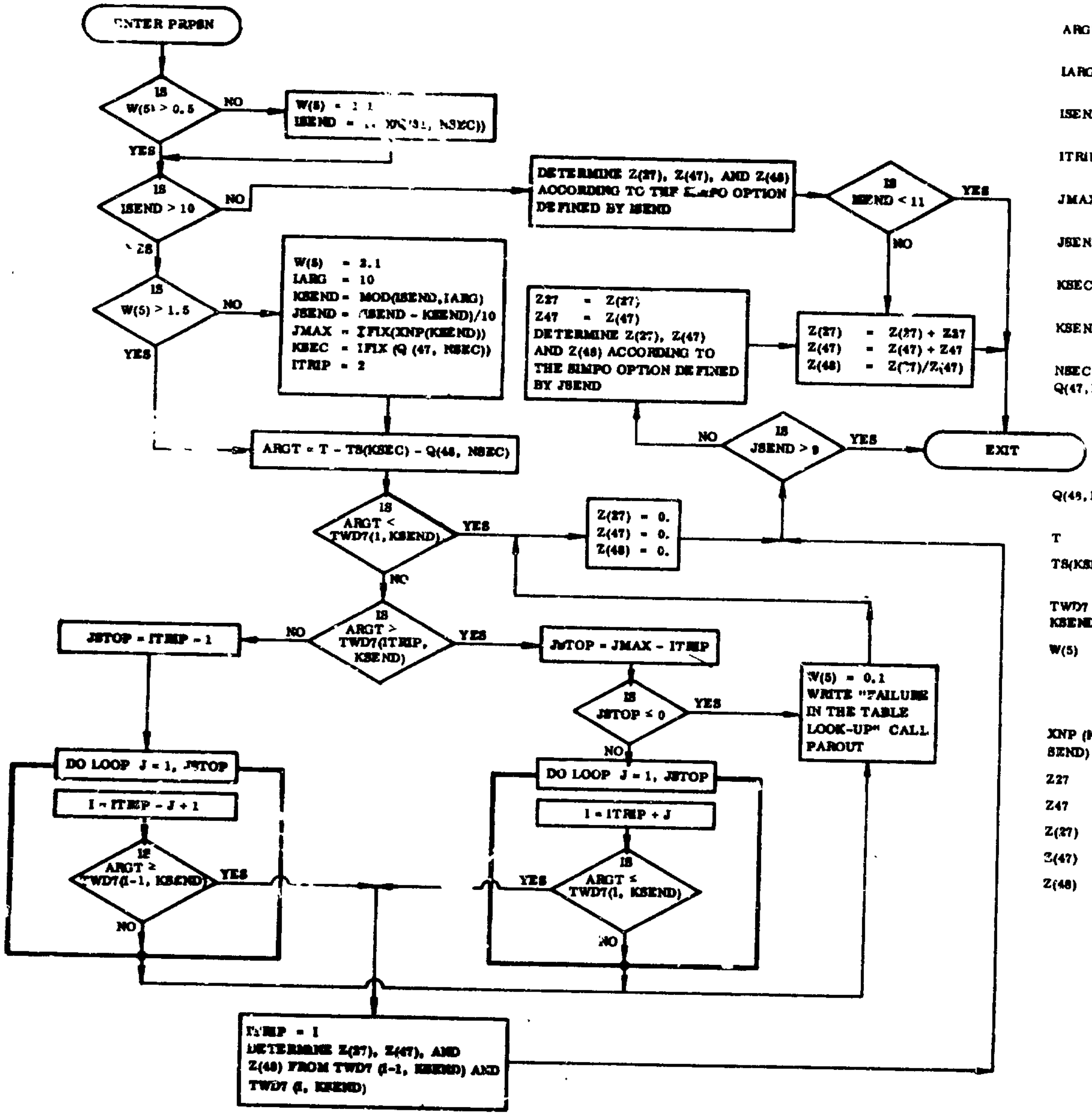
Figure 3-27. Subroutine AEROK Flow Diagram

3.2.3.13 PRPSN. The current total effective thrust, propellant flow rate, and specific impulse are determined in subroutine PRPSN. Three principal methods are available; these are use of a SIMPO* propulsion model, a thrust-time table, or a combination of SIMPO plus a thrust-time table. A summary flow diagram of this subroutine appears in Figure 3-28 and a FORTRAN listing is presented in Appendix IV-29.

If a SIMPO model is to be used by itself (e.g., with no thrust table) the current thrust level, propellant flow rate, and specific impulse are evaluated according to the option which is specified by program input. These options are defined in detail in Section 4.2.3.6 Propulsion Models. Upon completion of this evaluation, control is returned to subroutine EQTNL.

Whenever a thrust table is specified, the thrust and propellant flow rate elements which are defined by table are processed first. There are five propulsion tables which are available, any one of which can be specified during a simulation section. These tables utilize the same efficient memory procedure which is used in the atmosphere table of subroutine, except here the elements which have different definitions are dimensioned for five tables. These tables define vacuum thrust and propellant flow rate (if specified) as a function of a time argument which is equivalent to time measured from a reference time in a reference simulation section. Whenever the current time precedes the first table time argument, the vacuum thrust and propellant flow rate which are defined by the table are assumed to be zero. Upon definition of the vacuum thrust, the propellant flow rate can either be defined by its own table (correspondingly, five propellant flow rate tables are available which utilize the same time argument values as their associated thrust table) or from an effective vacuum specific impulse corrected to include the effects of all weight which is expended during the motor burn (e.g., expended inerts, thrust vector control fluids, etc.). The current thrust level is then determined as a function of atmospheric pressure, motor exit area, the number of motors, and the nozzle cant angle. See Section 4.6.5 for a complete definition of the available options. If no SIMPO model is to be added to the thrust table model, the effective specific impulse is then computed and control is returned to subroutine EQTNL.

*A SIMPO propulsion model consists of three propulsion parameters from which the effective thrust, propellant flow rate, and specific impulse can be obtained.



ARGT	IS THE CURRENT TIME ARGUMENT TO BE USED IN THE TABLE LOOK-UP PROCEDURE
IARG	IS THE BASE FOR THE MODULUS FUNCTION WHICH IS USED TO DEFINE KSEND
ISEND	DEFINES WHICH PROPULSION OPTION AND/OR TABLE IS TO BE USED DURING THE CURRENT SIMULATION SECTION
ITRIP	IS THE MEMORY FLAG WHICH IS USED TO DETERMINE WHERE TO INITIATE THE TABLE SEARCH PROCEDURE
JMAX	IS THE TABLE SEARCH STOP PARAMETER WHICH LIMITS THE LENGTH OF THE TABLE SEARCH PROCEDURE
JSEND	DEFINES THE SIMPO PROPULSION OPTION TO BE COMBINED WITH THE SELECTED PROPULSION TABLE
KSEC	IS THE NUMBER OF THE SIMULATION SECTION FROM WHICH THE TIME ARGUMENT OF THE PROPULSION TABLE IS REFERENCED
KSEND	DEFINES THE PROPULSION TABLE TO BE USED DURING THE CURRENT SIMULATION SECTION
NSEC	IS THE NUMBER OF THE CURRENT SIMULATION SECTION
Q(47, NSEC)	IS THE SIMULATION SECTION NUMBER WHICH DEFINES THE SIMULATION SECTION TO WHICH THE TIME ARGUMENT OF THE PROPULSION TABLE IS REFERENCED DURING SIMULATION SECTION NSEC
Q(48, NSEC)	IS A CONSTANT TIME TERM WHICH IS USED WHEN DEFINING ARGT FOR THE NSEC-th SIMULATION SECTION
T	IS THE CURRENT ABSOLUTE TIME
TS(KSEC)	IS THE ABSOLUTE TIME AT THE INITIATION OF THE KSEC-th SIMULATION SECTION
TWD7 (I, KSEND)	IS THE I-th INPUT TIME ARGUMENT FOR THE KSEND-th PROPULSION TABLE
W(5)	IS THE INITIALIZATION FLAG W(5) = 0.1 FOR THE INITIAL CALL TO PRPSN DURING A SIMULATION SECTION W(5) = 1.1 FOR SUBSEQUENT CALLS TO PRPSN
XNP (KSEND)	IS THE NUMBER OF INPUT TIME ARGUMENTS (TWD7 (I, KSEND)) FOR THE KSEND-th PROPULSION TABLE
Z27	TEMPORARY VALUE OF Z(27) WHICH IS OBTAINED BY TABLE LOOK-UP
Z47	TEMPORARY VALUE OF Z(47) WHICH IS OBTAINED BY TABLE LOOK-UP
Z(27)	CURRENT TOTAL THRUST
Z(47)	CURRENT TOTAL TIME RATE OF CHANGE OF WEIGHT
Z(48)	CURRENT SPECIFIC IMPULSE

Figure 3-28. Subroutine PRPSN Flow Diagram

If a SIMPO model is to be added to a table model, this is accomplished after the effective thrust and propellant flow rates modeled by the thrust table have been computed. These parameters are then stored as Z27 and Z47 (temporary storage not to be confused with the Z array), respectively, and then the specified SIMPO model is employed to define the effective thrust and propellant flow rate which is modeled by SIMPO, according to the SIMPO option which is specified by input. These parameters are each added to Z27 and Z47, respectively, to obtain to total combined effective thrust level and propellant flow rate. The total combined specific impulse is then determined and control is returned to subroutine EQTNL.

3.2.3.14 TRAN1, TRAN2, AND TRAN3. Subroutines TRAN1, TRAN2, and TRAN3 simply execute the arithmetic operations required for the transformations defined by Eq. [M-5] or [M-7], and [M-8], respectively. The required initial vector components and the sine and cosine values are introduced to the program by means of the CALL statement list, thus providing generality of application to other matrices having the same form and eliminating redundant calculation of the sine and cosine values. The components of the transformed vector are returned to the calling subroutine (in this case, EQTNL) via the CALL statement. Listings of the TRAN1, TRAN2, and TRAN3 subroutines are presented in Appendix IV-30.

3.2.3.15 QUAD. Subroutine QUAD is used to constrain angular quantities to the principal cycle, that is, between 0 and 360 degrees. This is accomplished by repetitive addition or subtraction of 360 degrees to the angular quantity until the quantity is adjusted to be within the principal cycle. a FORTRAN listing of QUAD is presented in Appendix IV-31.

3.2.3.16 QARG. The purpose of subroutine QARG is to provide an automatic means to obtain either degrees or radians when evaluating the arcsine, arccosine, and arctangent functions and to provide the appropriate logic to bypass computational errors which occur when the argument of either the arcsine or arccosine function is outside the acceptable bounds (± 1). In this case the function is evaluated for the boundary value nearest the value of the argument. A FORTRAN listing is presented in Appendix IV-32.

3.2.4 WTVOL OVERLAY

The weight/volume overlay consists of a main driver program (WTVOL), the three subroutines required by the driver for the weight/volume computations (WTSCH, SOLVE, TAMPER), and six subroutines (PRINTW, PRINTV, PROTHR, PRWTSM, PRITEQ, PRITVA) used to print the weight/volume data. These ten routines are discussed in the subparagraphs of this section.

3.2.4.1 WTVOL Program. The WTVOL routine, an executive driver, controls the computation and printing of the weight and volume data and contains the logic for the four sizing options, iterating to: (1) gross weight at lift -off, (2) gross orbiter weight, (3) orbiter propellant weight, and (4) thrust-to-weight ratio at lift-off. A rapid convergence technique, Newton-Raphson, is used for the first three options. A modified straight iterative is used for the thrust-to-weight ratio at lift-off option. A flowchart for WTVOL is shown in Figure 3-29 and the FORTRAN listing is in Appendix IV-33.

3.2.4.2 WTSCH Subroutine. The subroutine WTSCH computes the propellant and design weights, vehicle geometry data, vehicle structure and subsystems weights, and sums the vehicle weights. Volume II of this report contains the explanation of the equations along with substantiating data curves. A flowchart for WTSCH is presented in Figure 3-30 and the FORTRAN listing appears in Appendix IV-34.

3.2.4.3 SOLVE Subroutine. The subroutine SOLVE employs a Newton-Raphson iteration procedure to drive the difference in gross stage weight that is the estimated input and the gross stage weight that is obtained from the WTSCH subroutine to zero. Appendix IV-35 contains the FORTRAN listing of SOLVE.

3.2.4.4 TAMPER Subroutine. The subroutine TAMPER acts as an interface for the WTVOL and GTSM overlays. Weights, propulsion characteristics (sea level and vacuum thrust and specific impulse), and vehicle reference areas determined in the WTSCH subroutine are defined for GTSM variable names. A pitch rate algorithm is evaluated. Two alternate entries (ORBSUM and BOOSUM) are used to store weights, volumes, and geometry which will be printed by the routine SUMOUT. A flow diagram for TAMPER is shown in Figure 3-31 and the FORTRAN listing is in Appendix IV-36.

3.2.4.5 PRINTW Subroutine. The weight breakdown for both the orbiter and booster stages is printed by the subroutine PRINTW. A sample output from this routine is shown in Sec. 5.1.1 . The driving program WTVOL calls PRINTW only if a converged case is being printed or if the input flag INTOUT has been properly set by the user. The FORTRAN listing for PRINTW is presented in Appendix IV-37.

3.2.4.6 PRINTV Subroutine. The volumes of the booster and orbiter stages are printed by the subroutine PRINTV which is called by WTVOL only for a converged case of if the input flag WTOUT has been properly set. A sample output printed by this routine is shown in Sec. 5.1.1 . The FORTRAN listing of the PRINTV routine appears in Appendix [illegible]

WTVOL SUBROUTINE FLOWCHART

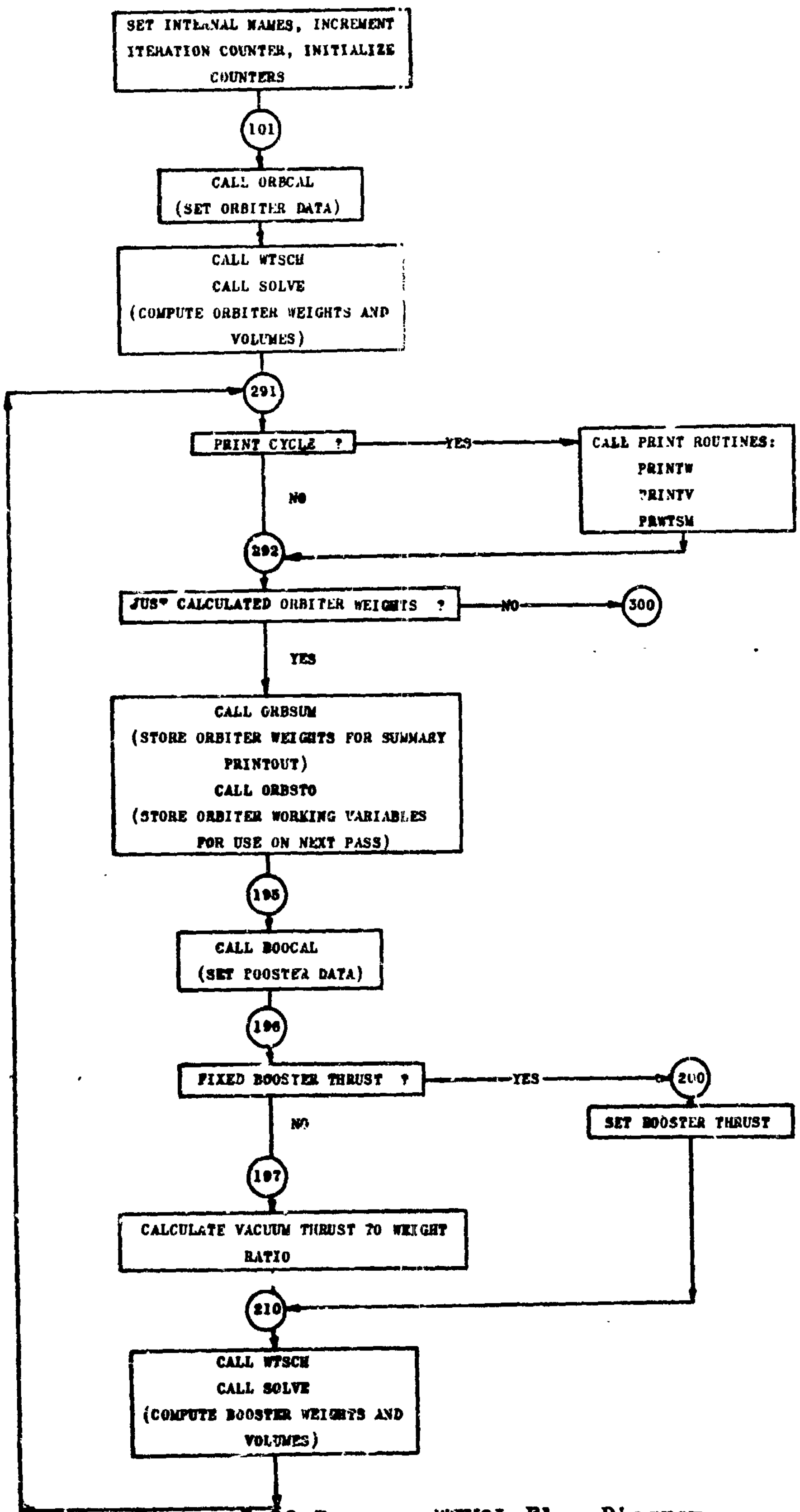


Figure 3-29 Program WTVOL Flow Diagram

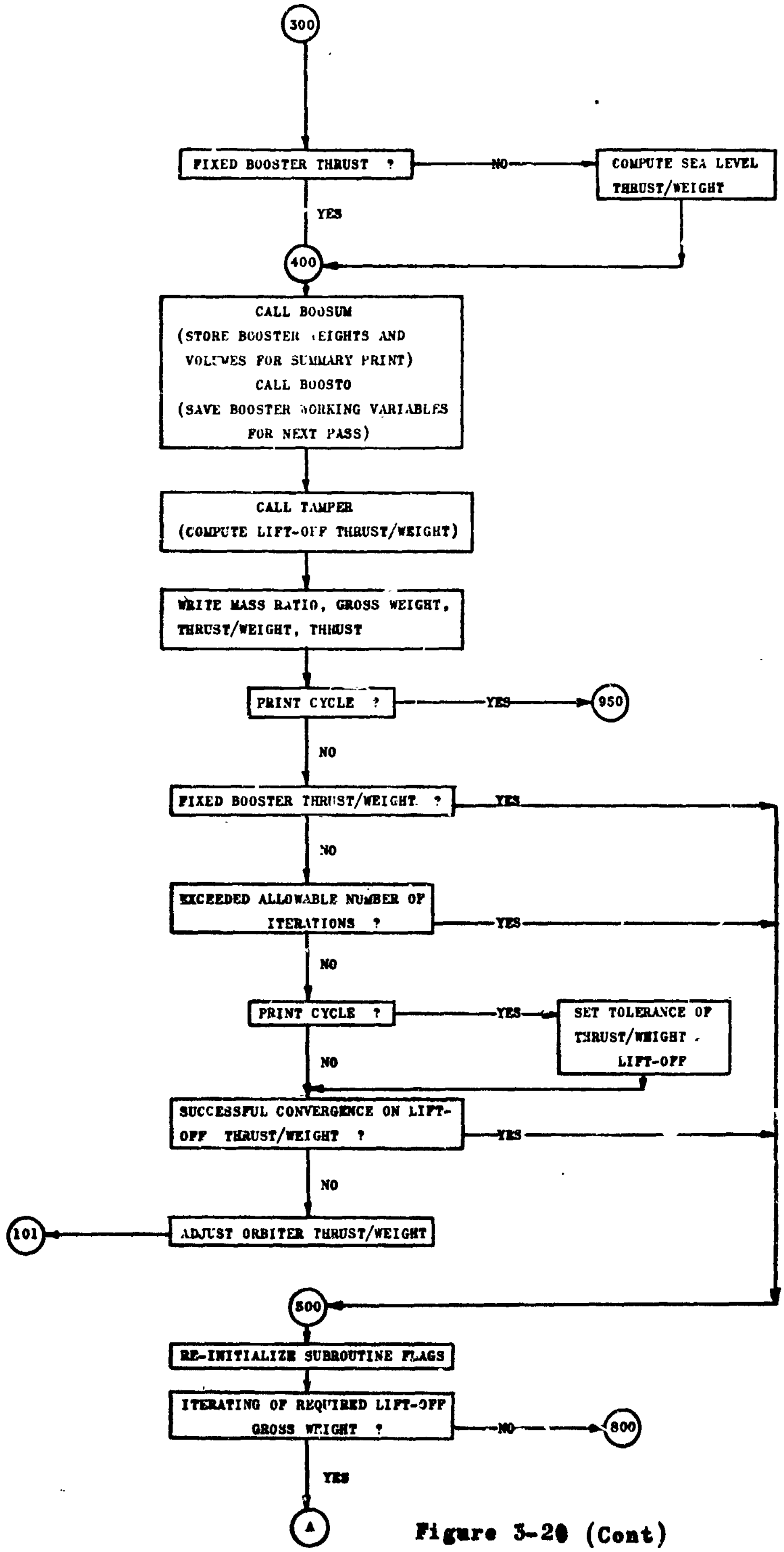


Figure 3-20 (Cont)

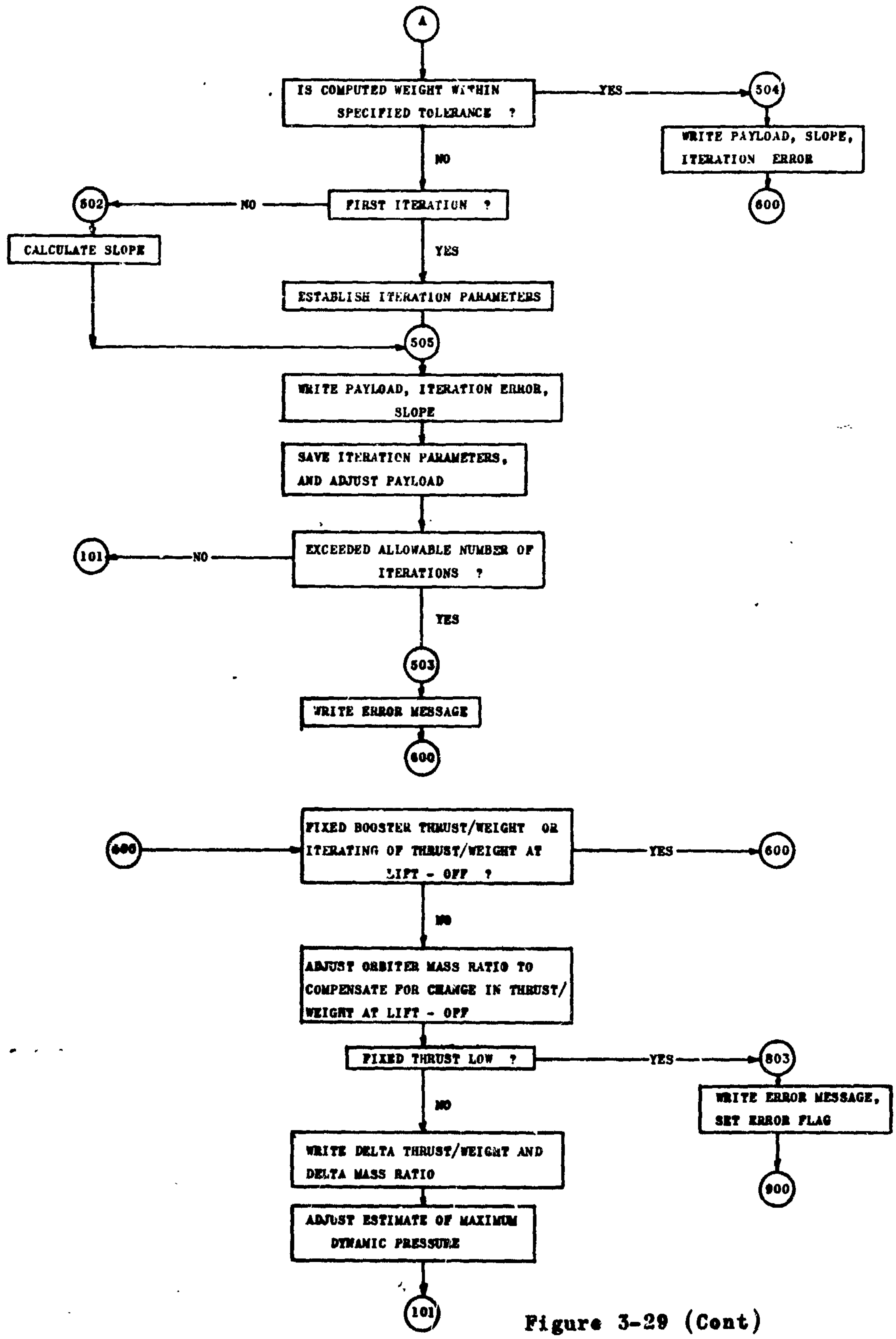


Figure 3-29 (Cont)

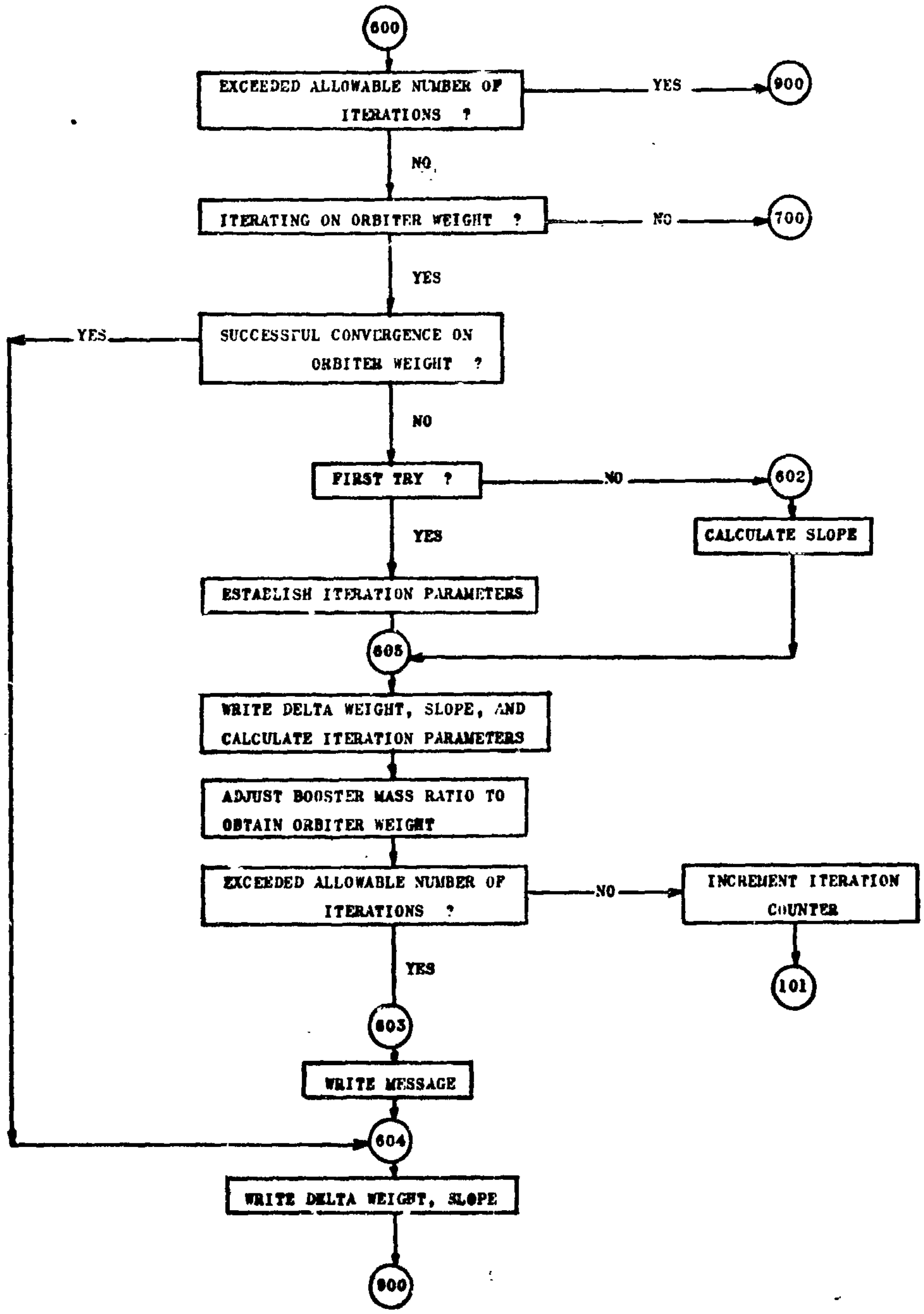


Figure 3-29 (Cont)

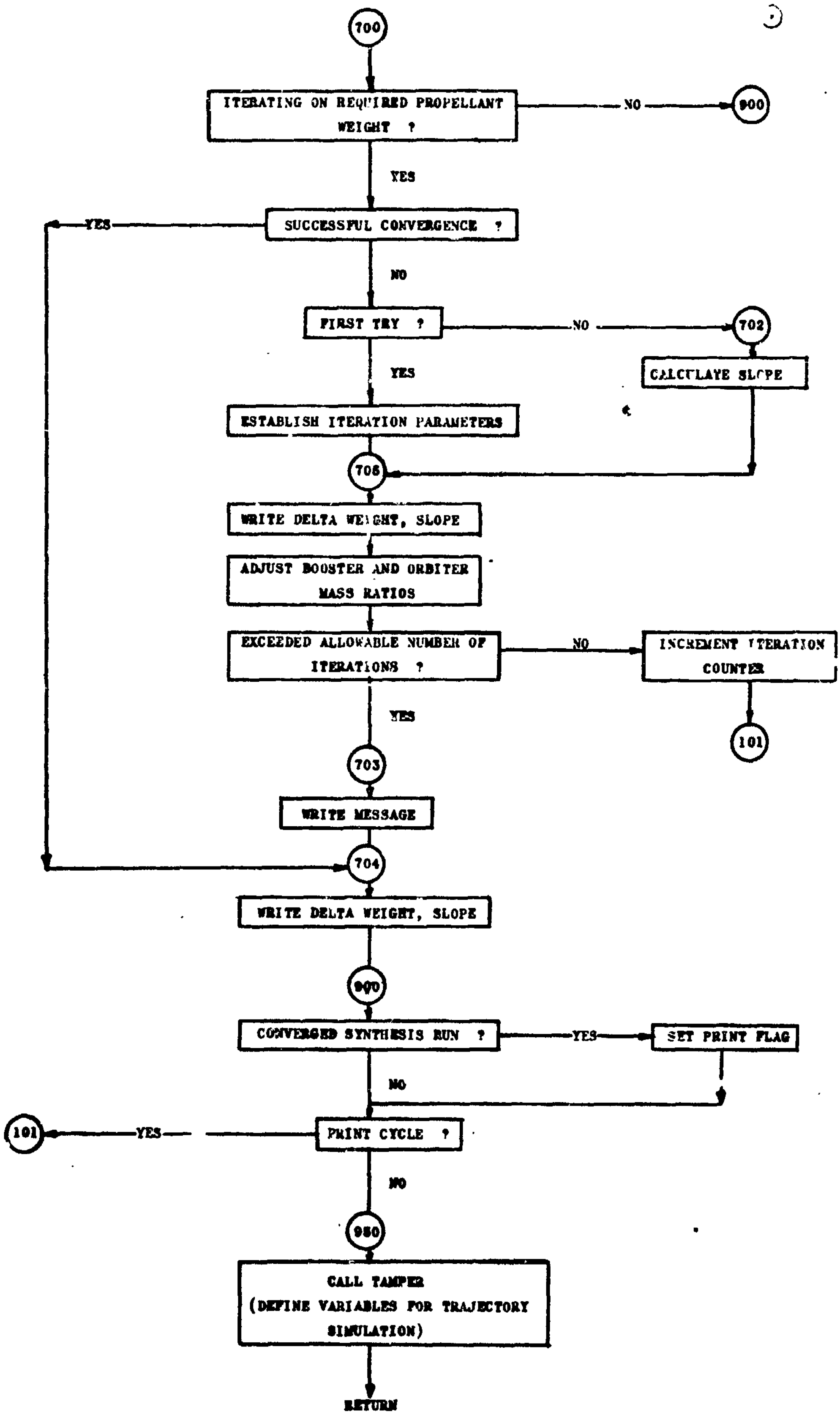


Figure 3-29 (Cont)

WTSCH SUBROUTINE FLOWCHART

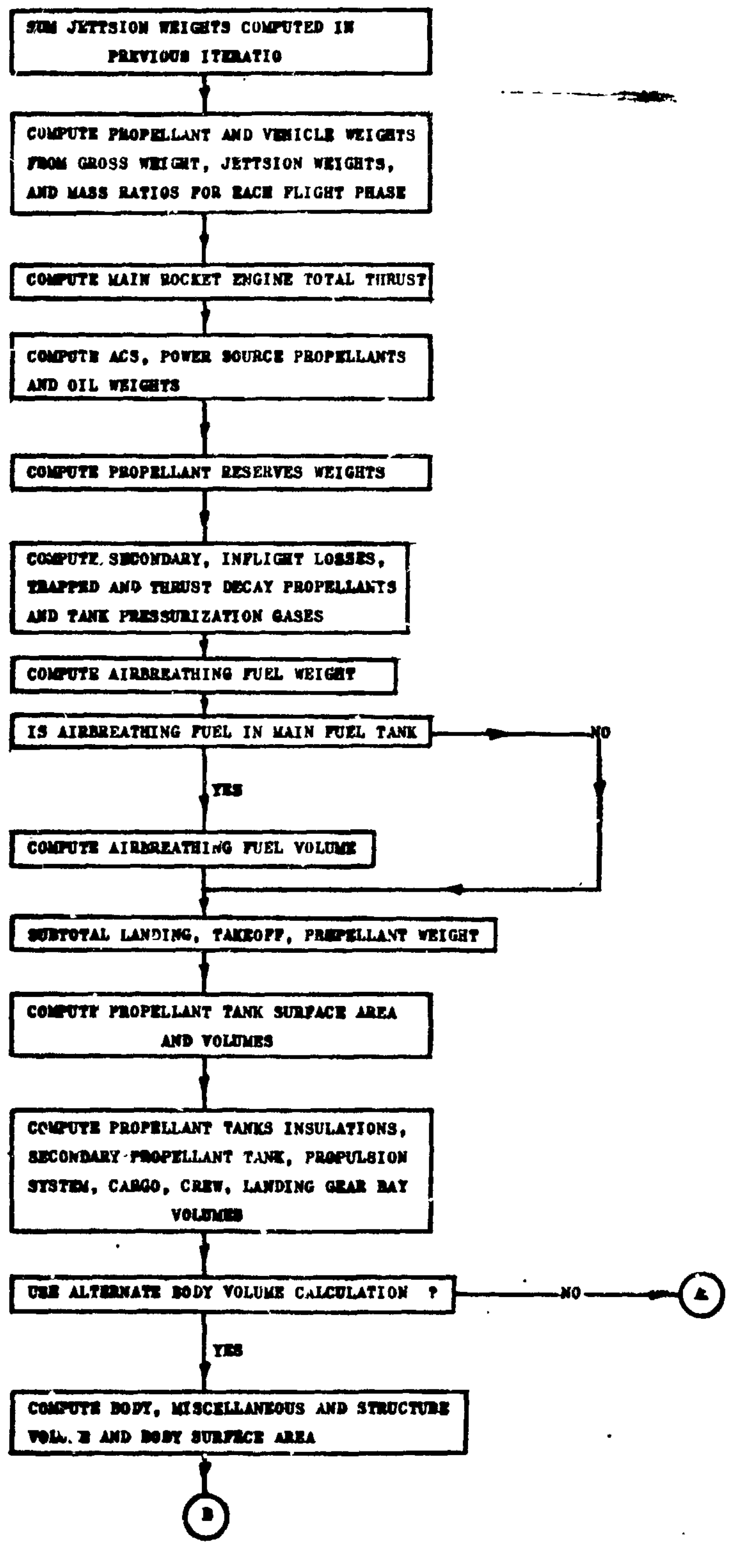


Figure 3-30 Subroutine WTSCH Flow Diagram

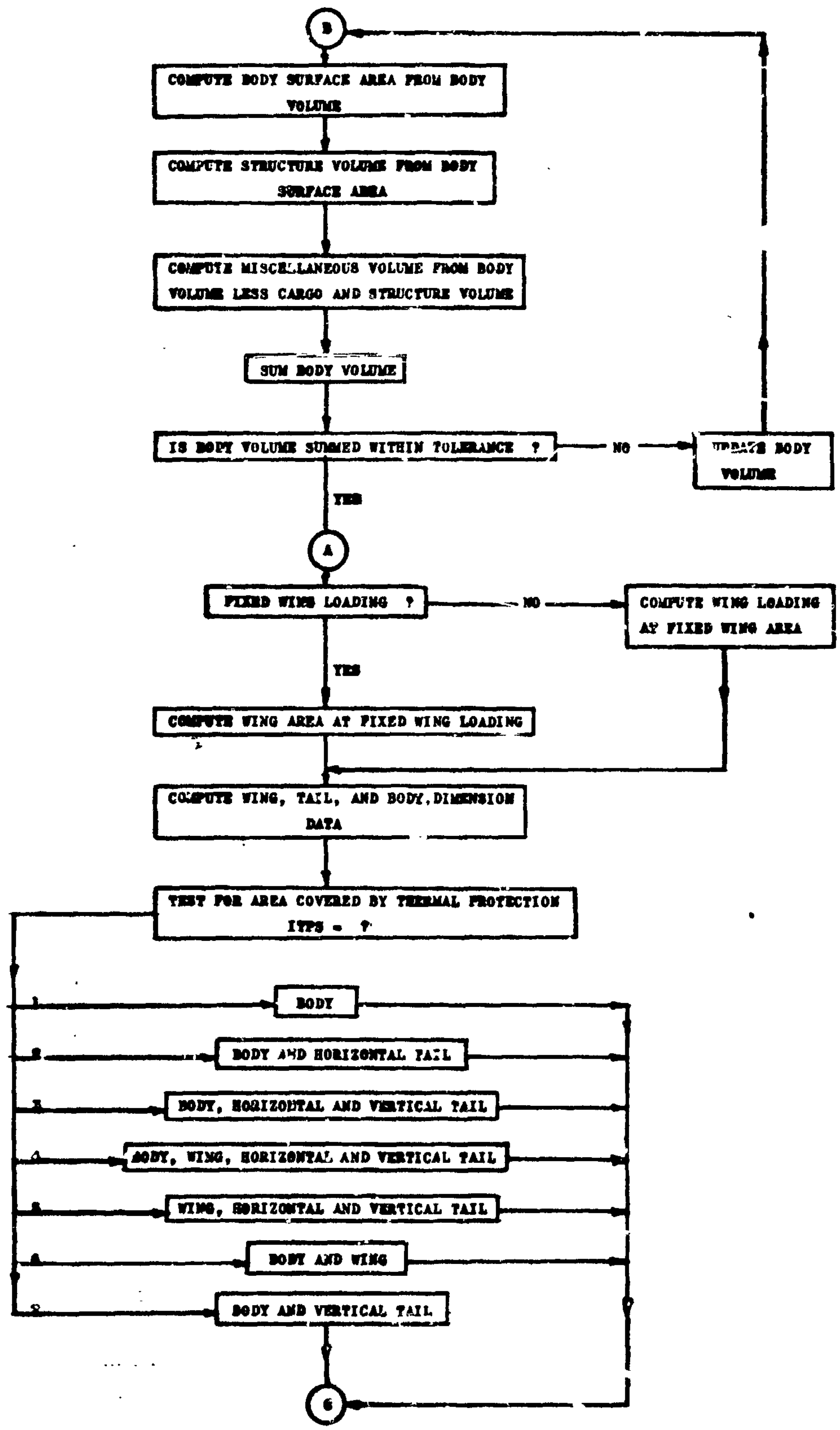


Figure 3-30 (Cont)

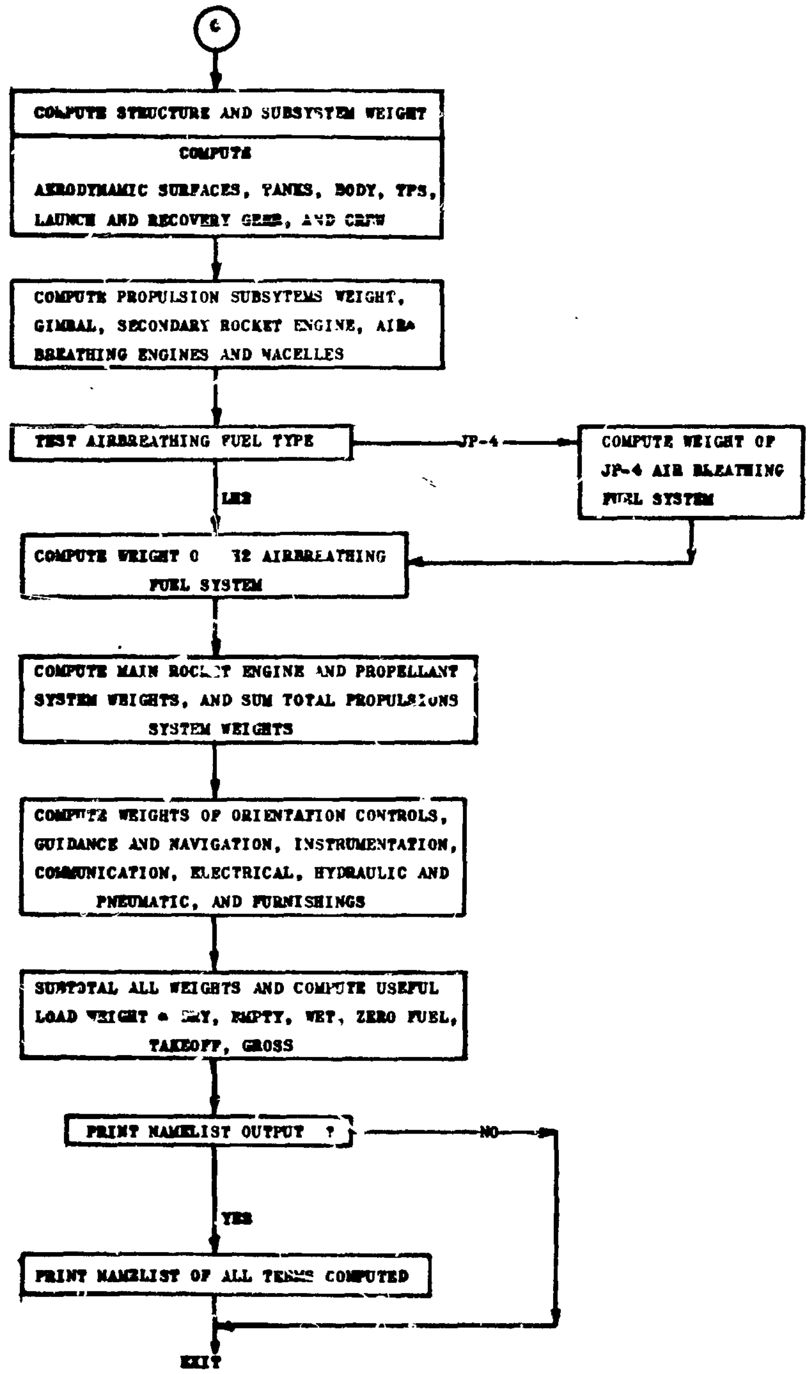


Figure 3-30 (Cont)

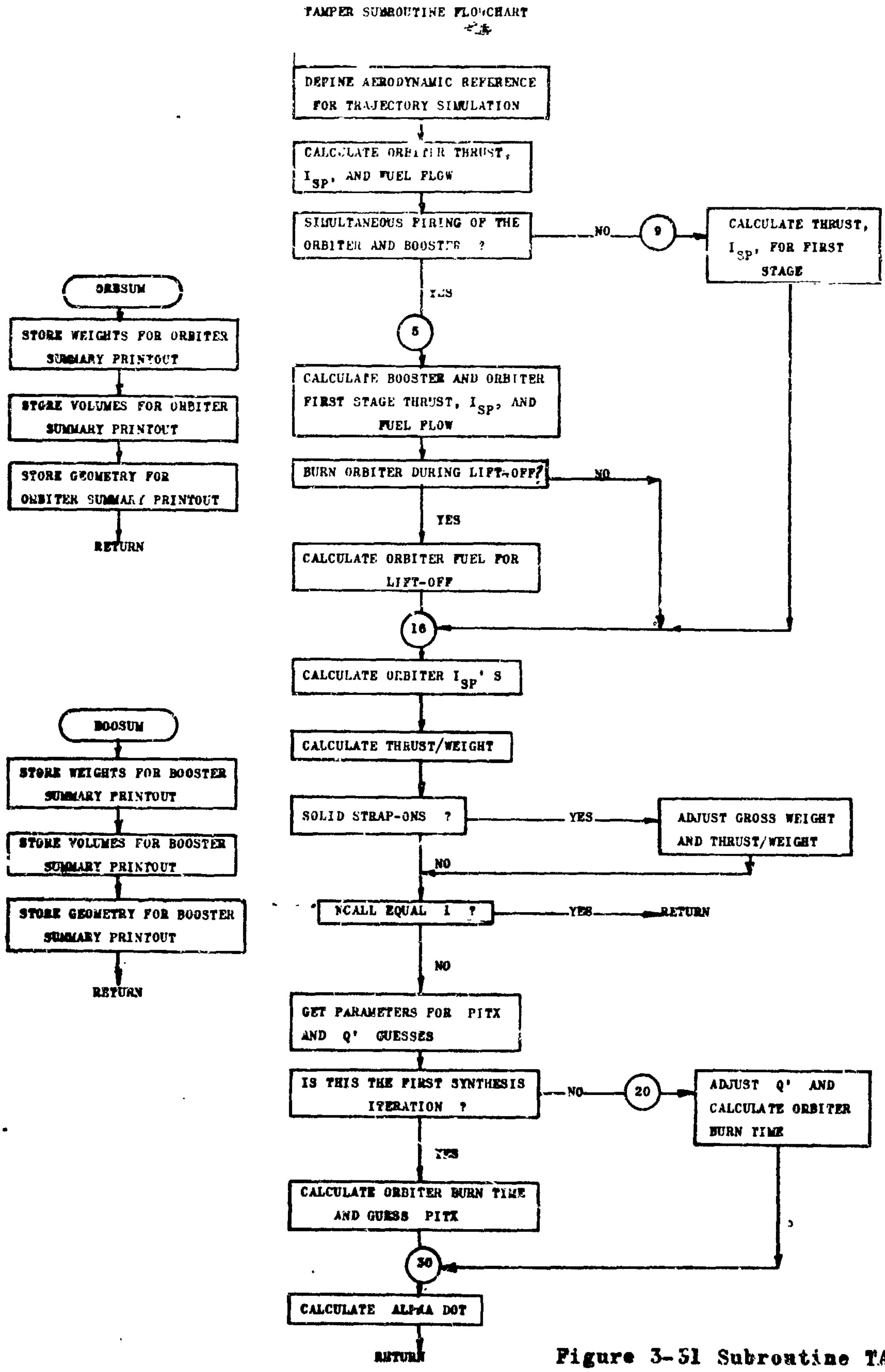


Figure 3-51 Subroutine TAMPER Flow Diagram

3.2.4.7 PROTHR Subroutine. The design data page of the sample output shown in Sec. 5.1.1 is printed by the subroutine PROTHR which is called from the subroutine PRINTV. This information is printed only for a converged case or if the print flag WTOUT has been set by the user. The FORTRAN listing of the subroutine PROTHR is shown in Appendix IV-39.

3.2.4.8 PRWTSM Subroutine. The weight summary information as shown in Sec 5.1.1 is printed by the subroutine PRWTSM which is called from WTVOL. The printout occurs only for a converged case or when the input flag WTOUT has been set by the user. Appendix IV-40 contains the FORTRAN listing for subroutine PRWTSM.

3.2.4.9 PRITEQ Subroutine. The input arrays of weight and volume coefficients that are non-zero are printed by the subroutine PRITEQ as shown in the sample output in Sec. 5.1.1 . Appendix IV-41 contains the FORTRAN listing of subroutine PRITEQ.

3.2.4.10 PRITVA Subroutine. The geometry sizing coefficients as shown in Sec. 5.1.1 are printed by the subroutine PRITVA which is called from the subroutine PRITEQ. The printout occurs for a converged case or if the flag WTOUT has been set by the user. A FORTRAN listing of PRITEQ is shown in Appendix IV-42.